ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

СКАНТЕК

Комплекс тестирования ECV

Версия 50.27

Февраль 2019

Россия, 119049, г. Москва, Донская ул., д. 15

Телефон: (499) 271-9661 • e-mail: 2b@scantech.ru

Оглавление

Глава
1

Основные понятия

Комплекс тестирования ECV (EMV Card Verification) предназначен для тестирования EMV-приложений на смарт-картах. ECV позволяет проверить полноту данных на карте и работоспособность карты при обслуживании транзакций, непротиворечивость и отсутствие избыточности данных, контролировать выполнение криптографических функций EMV-приложения, выявить причины сбоев в работе уже выпущенных карт, и ещё многое другое.

ECV – это эмулятор терминала в торговой точке (POS-терминала) с целым рядом дополнительных возможностей, которых нет в POS-терминале. EMV-приложение, тестируемое ECV, может быть размещено как на контактной карте (удовлетворяющей спецификациям ISO/IEC 7816), так и на бесконтактной карте или мобильном устройстве (взаимодействие с которыми осуществляется на основе протоколов, описанных в ISO 14443). Платежное приложение выбирает режим обработки транзакции (контактный или бесконтактный) автоматически в зависимости от метода использования носителя. Поэтому платежный аплет может быть размещен и на карте с двумя видами интерфейсов – контактным и бесконтактным (dual interface card). Мобильное устройство всегда осуществляет коммуникацию с POS-терминалом через NFC и эмулирует работу бесконтактной карты.

Хотя ECV эмулирует обработку транзакции в контактном и бесконтактном режимах, но между этими режимами существует большая разница. В контактном режиме любое EMV-приложение работает в соответствии со спецификациями, определенными в EMV. Integrated Circuit Card Specifications for Payment Systems. Book 1-4, а в бесконтактном – по спецификациям EMV Contactless Specifications for Payment Systems. Это связано с тем, что в бесконтактном режиме из-за определенных ограничений нельзя полностью поддержать тот же алгоритм обработки транзакции, что в контактном режиме.

Для бесконтактного режима нет единой спецификации. Каждая платежная система (например, Visa, MasterCard, …) реализовала собственный алгоритм обработки в бесконтактном режиме, и спецификации EMV Contactless Specifications for Payment Systems – это не более, чем набор общих положений об обработке бесконтактной транзакции в POS-терминале. Для каждой

платежной системы вводится понятие ядра (Kernel), которое отвечает за обработку бесконтактной транзакции для этой (и только этой) системы. Сведений, приведенных в спецификациях EMV Contactless Specifications, обычно недостаточно для реализации ядра обработки бесконтактной транзакции в терминале. Кроме открытых спецификаций EMV Contactless Specifications нужны ещё спецификации работы платежного приложения в бесконтактном режиме для конкретной платежной системы. А эти спецификации предоставляются только партнерам платежной системы (проще говоря, покупаются за немалые деньги).

В связи с этим основная разница обработки контактной и бесконтактной транзакции в ECV состоит в следующем.

- Обработка контактной транзакции регламентируется строгими спецификациями EMV. Integrated Circuit Card Specifications for Payment Systems, в связи с чем этот процесс полностью определен и может быть выполнен эмулятором терминала для EMV-приложения любой платежной системы.
- Обработка бесконтактной транзакции описывается общими спецификациями EMV Contactless Specifications и зависит от платежной системы. Эмулятор терминала выполняет обработку бесконтактной транзакции не для всех, а только для некоторых платежных систем.

Перед тем как начать использовать комплекс тестирования ECV, рекомендуется ознакомиться со следующими понятиями:

- формат объектов данных, которыми обмениваются карта и терминал
- основные этапы выполнения транзакции в контактном режиме
- особенности выполнения транзакции в бесконтактном режиме.

Краткое объяснение этих сущностей приведено в последующих разделах главы. Если вы хорошо знакомы с рассматриваемым вопросом, можете перейти к следующей главе.

Элементы и объекты данных

Любое EMV-приложение использует некоторый набор элементов данных. Элементы данных являются логическими структурами, которые при необходимости (например, при передаче данных в терминал) отображаются в объекты данных. Существуют различные способы отображения элементов данных в объекты данных. В спецификациях EMV используется кодирование BER-TLV, формализованное в стандарте ISO/IEC 8825. В соответствии с ISO/IEC 8825 любой объект данных определяется двумя или тремя полями: тэгом (Tag), длиной (Length) и значением (Value). Поле значения должно быть опущено, если длина равна 0. Поле тэга состоит из одного или более байтов (в спецификациях EMV – один или два байта). Структура поля тэга показана в следующих таблицах.

Структура первого байта поля тэга.

8	7	6	5	4	3	2	1	Значение
x	x							Класс объекта данных
		0						Примитивный объект
		1						Составной объект
			1	1	1	1	1	Имеются другие байты тэга
			x	x	x	x	x	Номер тэга

Структура последующих байтов поля тэга.

8	7	6	5	4	3	2	1	Значение
1								Имеются другие байты тэга
0								Последний байт поля тэга
	x	x	x	x	x	x	x	Номер тэга (больше 0)

Два первых бита первого байта поля тэга определяют класс объекта данных следующим образом:

- 00 – универсальный класс
- 01 – прикладной класс, который содержит объекты, относящиеся к определенной индустрии (например, индустрии расчетов по карточкам)
- 10 – контекстно-определенный класс (содержит объекты, определенные для некоторого стандарта)
- 11 – частный класс, содержащий объекты, которые определены конкретными спецификациями

Для группирования данных используются составные объекты данных, которые могут содержать другие составные объекты и примитивные объекты.

В спецификациях EMV составной объект данных называется template (например, FCI Template).

Поле длины объекта данных определяет количество байт в поле значения объекта. В стандарте EMV поле длины задается одним, двумя или тремя байтами. Если старший бит самого левого байта поля длины равен 0, то поле длины занимает один байт и определяет длину значения от 0 до 127. Если старший бит равен 1, то последующие биты определяют количество дополнительных байтов, используемых для представления поля длины.

Описанный выше метод кодирования данных BER-TLV позволяет для конкретного объекта данных однозначно определить его идентификатор, размер и значение. Таким образом, кодировка BER-TLV является универсальным средством представления данных.

Списки объектов данных

Для выполнения ряда команд карте требуются параметры транзакции и возможности терминала. Для большинства EMV-приложений такие данные требуются для первой и второй команд GENERATE AC, а также для команды GET PROCESSING OPTIONS. Список объектов данных, необходимых для выполнения команды, в общем случае называется Data Object List (DOL). Для конкретных команд названия списков уточняются. Например:

- Card Risk Management Data Object List 1 (CDOL1) – список объектов данных для первой команды GENERATE AC
- Card Risk Management Data Object List 2 (CDOL2) – список объектов данных для второй команды GENERATE AC
- Processing Options Data Object List (PDOL) – список объектов данных для команды GET PROCESSING OPTIONS

Любой список объектов данных представляет собой перечень тэгов и длин объектов данных, которые должны быть переданы карте. Все списки записываются на карту в процессе её персонализации и считываются терминалом по команде READ RECORD. Терминал в соответствующих командах передает только значения объектов, перечисленных в списках.

Транзакция в контактном режиме

На рис. 1 показана общая схема обработки транзакции в контактном режиме. Приведенная диаграмма не отображает все особенности карточной операции в терминале, но позволяет понять, каким образом в этой операции участвует карта (платежное приложение). Кроме того, в нижеследующем описании обработки транзакции в контактном режиме отсутствуют важные детали, которые опущены, поскольку целью документа является не определение процесса обработки транзакции, а описание процесса взаимодействия терминала и карты.

Платежная операция начинается с выбора терминалом платежного приложения на карте. Для выполнения транзакции необходимо, чтобы терминал поддерживал платежное приложение, которое находится на карте. На этапе выбора приложения терминал проверяет факт наличия такого приложения (если на карте несколько приложений, поддерживаемых терминалом, то терминал и, возможно, владелец карты выбирают, какое из них следует использовать для выполнения текущей транзакции).

Существует два способа выбора приложения: выбор через PSE (Payment System Environment) и прямой выбор. PSE – это список с идентификатором 1PAY.SYS.DDF01, в котором перечислены все платежные приложения на карте. Но PSE не является обязательным. Поэтому терминал может отыскать на карте все приложения, которые он обрабатывает, и составить свой список кандидатов для обработки. В любом случае для выбора обрабатываемого приложения используется команда SELECT, в результате выполнения которой терминалу передаются некоторые данные о выбранном платежном приложении.[1]

После того как приложение выбрано, терминал инициирует транзакцию. Для этого используется команда GET PROCESSING OPTIONS, с помощью которой терминал сообщает карте данные, необходимые ей для того, чтобы определиться с особенностями выполняемой операции.[2] В ответе на команду GET PROCESSING OPTIONS карта сообщает терминалу о своих возможностях по выполнения транзакции и требованиях к терминалу (через Application Interchange Profile – AIP) и предоставляет ссылки на данные, которые терминал должен прочитать, чтобы успешно выполнить транзакцию (определяются в Application File Locator – AFL).

---

[1] Эти данные называются File Control Information (FCI) и содержат идентификатор приложения (AID), метку приложения и предпочтительные языки общения терминала с владельцем карты.

[2] Данные, необходимые карте для инициирования транзакции, определяются в списке Processing Options Data Object List (PDOL), предоставляемом в ответ на команду SELECT. Если никакие данные от терминала не нужны, то список PDOL отсутствует, и с командой GET PROCESSING OPTIONS никакие данные не передаются.

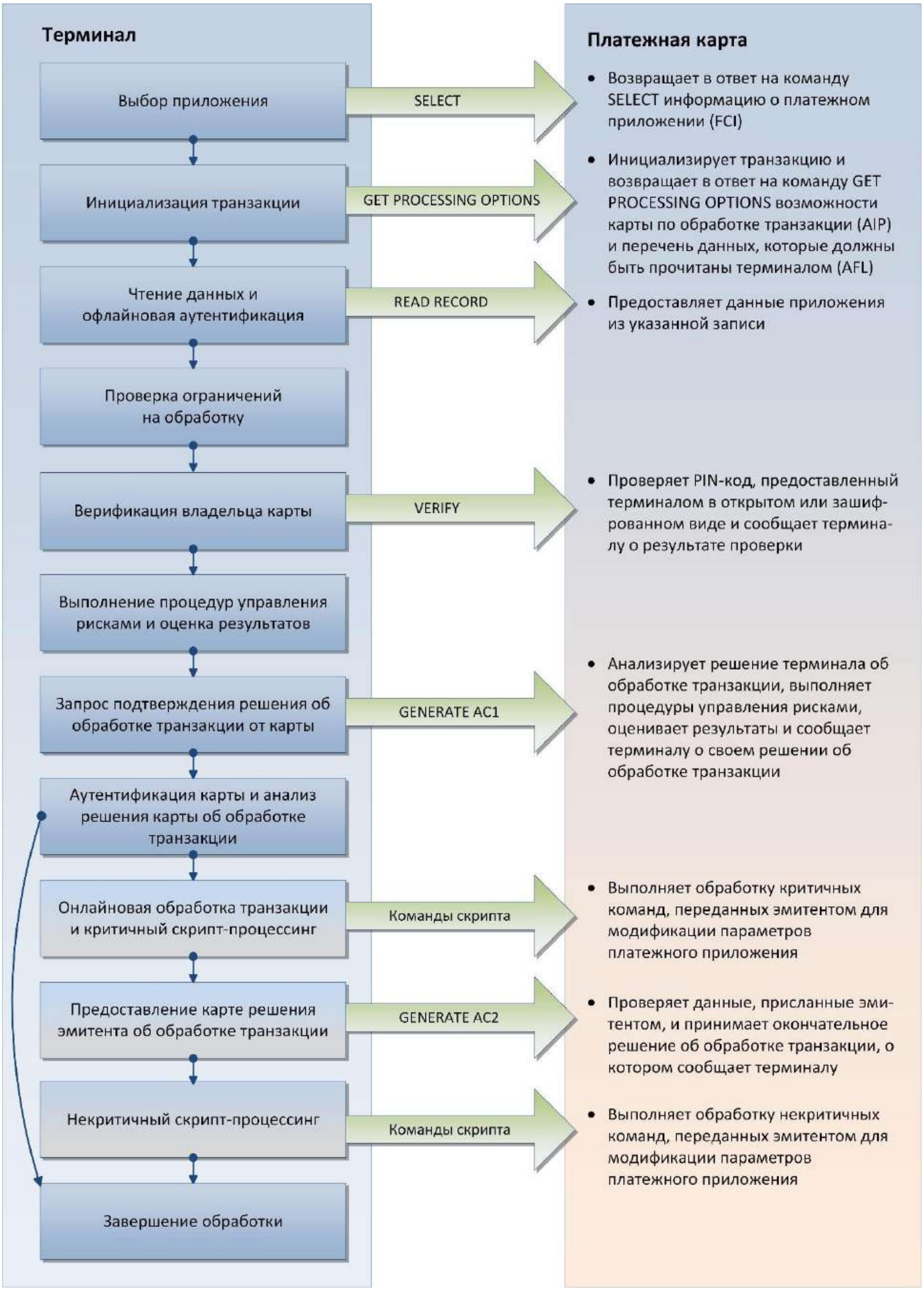


Рис. 1. Обработка транзакции в контактном режиме.

Терминал считывает все записи, указанные картой, с помощью команд READ RECORD и приступает к выполнению офлайновой аутентификации данных, предоставленных картой. В этот момент полностью может быть выполнена только аутентификация по методу SDA или DDA. Аутентификация данных по методу CDA (в силу особенностей реализации) выполняется полностью только после получения ответа от первой команды GENERATE AC.

Затем терминал приступает к выполнению процедур проверки ограничений по применению приложения (сверяются номера версий, проверяется срок действия карты и т. п.) и контролю стоп-листов.

Терминал просматривает список методов верификации владельца карты, предоставленный картой, выбирает подходящий метод и верифицирует владельца карты.

Терминал по указанию карты может также выполнить процедуры управления рисками, контролирующие сумму и количество последовательных транзакций, выполненных в offline.

Далее терминал производит анализ результатов выполненных им проверок с использованием критериев, сформулированных терминалу обслуживающим банком и эмитентом карты. В результате терминал принимает одно из трех решений об обработке транзакции:

1. Одобрить транзакцию в офлайновом режиме.

2. Передать транзакцию на авторизацию эмитенту карты.

3. Отвергнуть транзакцию в офлайновом режиме.

О своём решении терминал сообщает карте с помощью первой команды GENERATE AC. Вместе с командой GENERATE AC карте передаются данные транзакции, результаты процедуры управления рисками терминала и реквизиты терминала. На основе полученных данных карта выполняет собственные процедуры управления рисками и производит анализ результатов выполненных ею проверок с использованием критериев, сформулированных эмитентом карты. В результате карта принимает собственное решение об обработке транзакции – одно из трех решений, описанных выше, но ранг решения карты всегда не ниже ранга решения терминала (например, если терминал принял решение отвергнуть транзакцию, то карта не имеет права изменить это решение). В EMV эта зависимость является фундаментальной и контролируется как картой, так и терминалом.

Если карта принимает решение о том, что транзакция должна быть отправлена на авторизацию эмитенту, то карта формирует специальную криптограмму, представляющую собой подпись реквизитов транзакции, карты и терминала. Во всех остальных случаях формируются криптограммы,

информирующие о завершении транзакции (одобрении или отклонении транзакции в офлайновом режиме). Кроме того, в случае использования метода аутентификации CDA криптограмма заносится в специальный пакет данных, который зашифровывается на ключе карты и служит для офлайновой аутентификации данных терминалом.

Терминал, получив данные от карты по команде GENERATE AC, выполняет аутентификацию карты, если применяется метод аутентификации CDA (расшифровывает пакет данных с криптограммой и убеждается, что данные в пакете корректны). Затем терминал проверяет, какую криптограмму вернула карта. Если возвращена криптограмма одобрения или отклонения транзакции в офлайновом режиме, то обработка на этом завершается. Когда транзакция должна быть отправлена на авторизацию эмитенту, терминал отправляет (через хост обслуживающего банка) криптограмму и другие относящиеся к транзакции данные эмитенту.

Получив данные, эмитент авторизует транзакцию (проверяет криптограмму и некоторые другие параметры). По результатам этих проверок эмитент принимает решение о том, отклонить или одобрить транзакцию, и, в свою очередь, формирует криптограмму, которая используется картой для аутентификации эмитента. Ответ эмитента отправляется обслуживающему банку, который направляет его терминалу.

В ответе эмитента могут присутствовать команды скрипт-процессинга, предназначенные для карты. С помощью этих команд эмитент имеет возможность менять параметры платежного приложения, разблокировать или изменить PIN-код, заблокировать приложение карты. Эмитент может присвоить каждой команде скрипт-процессинга статус критичной (команда направляется карте сразу после получения её терминалом до выполнения проверки ответа эмитента) или некритичной (команда передается карте после выполнения проверки ответа эмитента).

Терминал, получив ответ эмитента, передает карте решение эмитента, криптограмму эмитента и уточненные данные своих проверок во второй команде GENERATE AC (или команде EXTERNAL AUTHENTICATE, если платежное приложение использует эту команду для проверки ответа эмитента). Карта аутентифицирует эмитента путем проверки криптограммы. Если аутентификация эмитента прошла успешно, то карта выполняет решение эмитента о завершении транзакции и формирует криптограмму одобрения или отклонения транзакции. Когда аутентификация эмитента провалилась, транзакция обычно отвергается.

Ниже приводится более подробное описание отдельных этапов обработки транзакции и объектов данных, которыми обмениваются терминал и карта.

Данные на карте

Данные, которые необходимы для выполнения транзакции, считываются терминалом из записей файлов платежного приложения по команде READ RECORD. Но не все данные, которые могут потребоваться терминалу, расположены в записях файлов. Некоторые данные хранятся в виде отдельных объектов и при необходимости терминал извлекает их с карты с помощью команды GET DATA.

Вопросы безопасности

Важнейшим свойством платежного приложения является использование криптографических функций для повышения безопасности финансовых операций. Основные задачи для повышения безопасности финансовых операций, решаемые приложением, следующие.

1. Обеспечение аутентификации приложения на карте (или аутентификации карты). Под аутентификацией карты понимается процесс доказательства её подлинности, т. е. того факта, что карта эмитирована банком, авторизованным на эмиссию карт. В случае офлайновой транзакции эмитент полностью делегирует карте функцию принятия решения по результату выполнения операции. Очевидно, что можно доверять только решениям карты, подлинность которой доказана. Поэтому так важна её надежная аутентификация. Аутентификация карты осуществляется терминалом и (или) эмитентом. В офлайновых операциях аутентификация карты производится только терминалом и называется офлайновой аутентификацией. В случае онлайновой транзакции аутентификация карты может осуществляться и терминалом, и эмитентом. Аутентификация карты, выполняемая эмитентом, называется онлайновой.

2. Обеспечение аутентификации эмитента и проверки целостности присылаемых им данных за счёт проверки подписей данных, формируемых эмитентом.

3. Гарантирование эмитенту невозможности отказа держателя его карты от результата выполненной операции. Это обеспечивается тем, что по каждой выполненной транзакции эмитент получает от карты в своё распоряжение криптограмму приложения, которая представляет собой подпись наиболее критичных данных транзакции. Соответствие криптограммы данным транзакции подтверждает факт её выполнения.

4. Проверка целостности обмена критическими данными между картой и терминалом.

5. Обеспечение конфиденциальности данных в информационном обмене между картой и эмитентом, картой и терминалом.

6. Предоставление механизма надежной верификации владельца карты с помощью офлайновой проверки PIN-кода.

Платежное приложение реализует функции обеспечения безопасности транзакции на используемых в стандарте EMV асимметричном алгоритме шифрования RSA и симметричном алгоритме шифрования DES.

Сначала рассмотрим алгоритм шифрования RSA. Если отвлечься от терминологии RSA, которая иногда может только запутать, то асимметричное шифрование основывается на том, что существуют два ключа – публичный и секретный. Если данные зашифрованы на публичном ключе, то они могут быть расшифрованы на секретном ключе и наоборот. Необходимость применения асимметричных алгоритмов шифрования в процедурах обеспечения безопасности вызвана тем, что терминал не должен обладать секретами карты (симметричное шифрование всегда подразумевает знание участниками информационного обмена общего секрета).

Для того чтобы терминал мог использовать асимметричные ключи карты для реализации функций обеспечения безопасности, необходимо создание PKI-инфраструктуры (PKI – Public Key Infrastructure). PKI-инфраструктура платежной системы в соответствии со стандартом EMV показана на рис. 2.

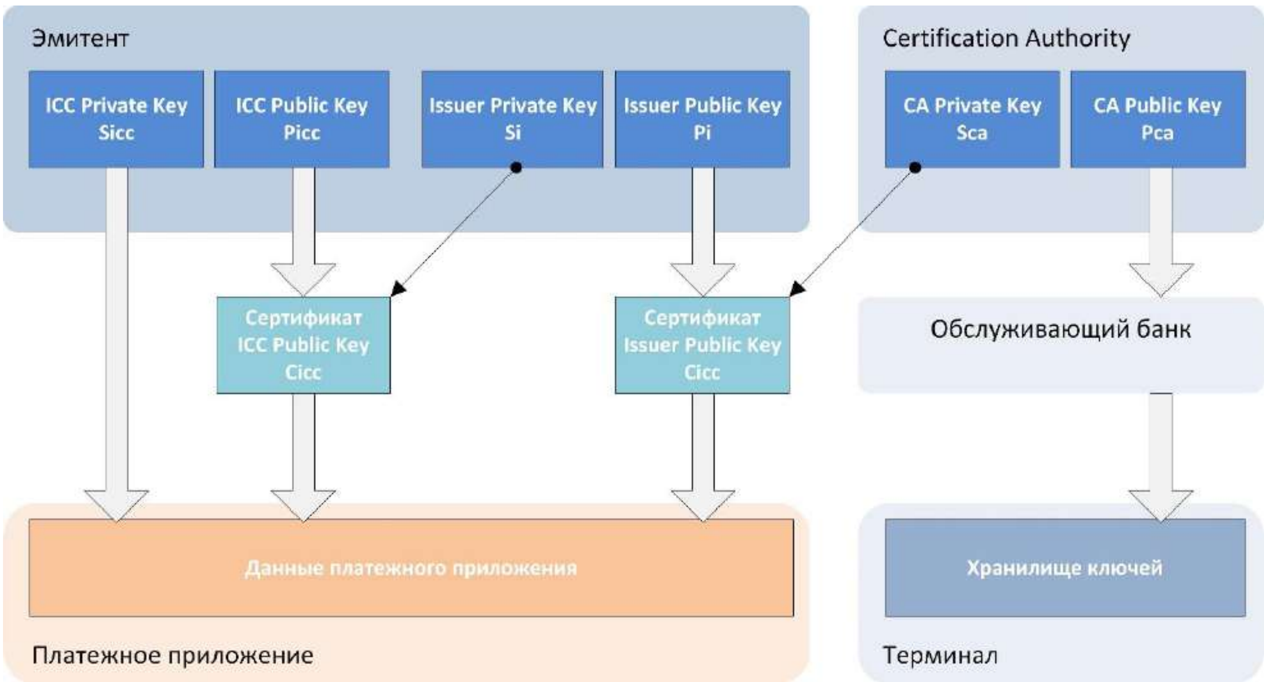


Рис. 2.PKI-инфраструктура платежной системы.

Корнем PKI-инфраструктуры является центр сертификации (Certification Authority – CA) платежной системы. Центр сертификации генерирует пары ключей RSA (открытых и секретных) и рассылает открытые ключи обслуживающим банкам для их загрузки в терминалы платежной системы. На втором

уровне дерева PKI-инфраструктуры находятся центры сертификации эмитентов-участников платежной системы. Эти центры также генерируют пары ключей RSA и получают в CA сертификаты открытых ключей. Сертификат открытого ключа представляет собой реквизиты открытого ключа (включая и сам открытый ключ), его срок действия, идентификатор эмитента и т. п., подписанные на секретном ключе центра сертификации платежной системы.[1]

Наконец, на нижнем уровне инфраструктуры располагаются открытые и секретные ключи карт эмитента. Открытые ключи карт подписываются в центре сертификации эмитента на секретном ключе центра сертификации эмитента.

В процессе персонализации платежного приложения на карту записываются сертификаты открытого ключа эмитента и открытого ключа карты, а также секретный ключ карты.

Для того чтобы получить открытый ключ карты, терминал считывает с карты сертификаты открытых ключей эмитента и карты. Далее с помощью открытого ключа CA, хранящегося в терминале, терминал проверяет правильность сертификата открытого ключа эмитента.[2]

После доказательства правильности сертификата открытого ключа эмитента терминал с помощью восстановленного открытого ключа эмитента проверяет правильность сертификата открытого ключа карты. Если сертификат корректен, то терминал получает доступ к открытому ключу карты. Открытый ключ карты используется терминалом для аутентификации приложения на карте и зашифрования конфиденциальных данных, пересылаемых приложению (например, PIN-кода). Аутентификация приложения осуществляется путем проверки подписи определенных данных, сформированной картой на секретном ключе карты. Если подпись карты верна, то это означает, что она была сделана секретным ключом карты, соответствующим её открытому ключу, сертифицированному эмитентом. Это в свою очередь означает, что карта была эмитирована банком-эмитентом, открытый ключ которого был сертифицирован центром сертификации платежной системы. Таким образом, доказывается факт выпуска карты банком, уполномоченным эмитировать карты платежной системы.

Для формирования криптограмм приложения, подписей данных, передаваемых между эмитентом и картой, а также для зашифрования обмена

---

[1] В терминологии EMV для формирования сертификатов используется функция Sign на секретном ключе. Эта функция зашифровывает сертификат на секретном ключе и он может быть расшифрован только на открытом ключе, соответствующем секретному ключу.

[2] Сертификат расшифровывается на публичном ключе. В терминологии EMV эта функция называется Recover. В результате расшифрования сертификата терминал получает доступ к открытому ключу эмитента, который хранится в сертификате. Но в сертификате может храниться не весь открытый ключ эмитента, а только его часть. Другая часть ключа извлекается из объекта данных, который называется Issuer Public Key Remainder.

конфиденциальными данными между ними, используется симметричный алгоритм 3DES (ISO 11568-2). В этом случае участники информационного обмена имеют общий секрет – секретный ключ 3DES. Вернее, набор секретных ключей, поскольку в целях безопасности платежное приложение и эмитент используют несколько ключей.

Справедливости ради нужно сказать, что набор секретных ключей в платежном приложении определяется разработчиком. Поэтому говорить о каких-то стандартах не приходится. Для примера можно привести, какие ключи используются в приложении, созданном по спецификациям EMV CCD (Common Core Definitions):

- ключ Ka (Application Cryptogram Key) – используется для вычисления криптограмм приложения
- ключ Ki (MAC Key) – применяется для вычисления подписи данных (Message Authentication Code – MAC) в командах скрипт-процессинга
- ключ Kc (Encipherment Key) – используется для зашифрования конфиденциальных данных в командах скрипт-процессинга

Эмитент хранит только мастер-ключи симметричного криптографического алгоритма, которые в процессе персонализации карты преобразуются (диверсифицируются) в уникальные ключи платежного приложения (диверсификация мастер-ключа эмитента может быть выполнена несколькими способами с использованием Application PAN и Application PAN Sequence Number). Таким образом, каждое платежное приложение использует для криптографических операций набор уникальных ключей. Более того, уникальный ключ платежного приложения диверсифицируется для каждой транзакции в сессионный ключ. Для получения сессионного ключа используется Application Transaction Counter (ATC) – счетчик, который увеличивается при выполнении каждой транзакции.

В результате для каждой транзакции карта и эмитент используют уникальный ключ симметричного криптографического алгоритма, что значительно повышает безопасность.

Верификация владельца карты

Верификация владельца карты выполняется для того, чтобы установить факт идентичности лица, получившего карту от её эмитента (клиента банка), с лицом, совершающим по карте операцию. Ключевым объектом данных для верификации владельца карты является объект Cardholder Verification Method (CVM) List, который хранится на карте. Этот объект определяет список методов и условий верификации владельца карты.

Одним из основных методов верификации владельца карты является процедура проверки PIN-кода. Проверка PIN-кода хотя и является только

одним из возможных методов верификации, но получили распространение в силу своей надёжности и эффективности.

Существует два различных метода офлайновой проверки PIN-кода:

- проверка PIN-кода, передаваемого на карту в открытом виде
- проверка PIN-кода, передаваемого на карту в зашифрованном виде

Офлайновая проверка PIN-кода всегда начинается с того, что терминал выдает команду GET DATA для получения объекта данных PIN Try Counter. Значение этого объекта – это количество оставшихся попыток ввода PIN-кода. Если у владельца карты еще остались попытки для ввода PIN-кода, то терминал получает от владельца карты значение его PIN-кода (от 4-х до 12-ти цифр) и формирует из PIN-кода PIN-блок в ISO Format 2, который представлен на рис. 3.



Рис. 3.PIN-блок в ISO Format 2.

Если в CVM указано, что PIN-код должен быть предъявлен в незашифрованном виде, то терминал передает на карту команду VERIFY, в поле данных которой содержится значение PIN-блока в ISO Format 2. Когда PIN-код должен быть предъявлен в зашифрованном виде, то сначала терминал получает от карты случайное число с помощью команды GET CHALLENGE, затем формирует сертификат PIN-кода, который содержит PIN-блок и полученное случайное число, и зашифровывает сертификат на открытом ключе карты. Зашифрованный сертификат PIN-кода подается на карту в качестве данных команды VERIFY. Карта сравнивает полученное значение PIN-кода со значением, хранимым на карте (предварительно расшифровывая сертификат PIN-кода на секретном ключе карты и проверяя сертификат, когда это необходимо). Если они совпадают, то проверка PIN-кода считается выполненной успешно. Каждый раз, когда предъявлен неправильный PIN-код, количество оставшихся попыток ввода PIN-кода уменьшается на 1 (при достижении нуля PIN-код будет заблокирован).

Онлайновая проверка PIN-кода никак не связана с картой. Устройство ввода PIN-кода запрашивает PIN-код от владельца карты и зашифровывает его по специальному алгоритму, обычно используя алгоритм DUKPT (Derived Unique Key Per Transaction). Верификация владельца карты терминалом на этом завершается, поскольку значение PIN-кода проверяет эмитент, а не карта.

Управление рисками терминала

Процедуры управления рисками, выполняемые терминалом, являются элементом обеспечения безопасности платежных операций и включаю в себя три механизма борьбы с карточным мошенничеством:

- контроль размера операций, выполненных по карте
- случайный выбор транзакции для её онлайновой авторизации эмитентом
- проверка офлайновой активности использования карты

По мере выполнения процедур аутентификации карты, проверки ограничений на обработку транзакции, верификации владельца карты, управления рисками терминал формирует поле с набором признаков, информирующих о результатах проверок, выполняемых терминалом в процессе обработки транзакции, которое называется Terminal Verification Results (TVR). После завершения всех процедур терминал выполняет анализ всех ситуаций, зафиксированных в TVR. Целью такого анализа является выработка терминалом рекомендательного решения о том, каким образом с точки зрения терминала должна быть продолжена обработка транзакции. Возможны три решения терминала:

- транзакция должна быть одобрена в офлайновом режиме
- транзакция должна быть направлена для авторизации эмитенту
- транзакция должна быть отклонена в офлайновом режиме

Когда говорят о решении «с точки зрения терминала», то имеют в виду, что в действительности решение формируется на основе правил, определенных обслуживающим банком (платежной системой) и эмитентом карты. Для этого в спецификациях EMV определены два множества объектов данных, которые называются Issuer Action Codes (IAC) и Terminal Action Codes (TAC). В свою очередь, каждое из этих множеств состоит из трех объектов с суффиксами Denial, Online и Default. Таким образом, используются следующие объекты.

IAC-Denial	TAC-Denial
IAC-Online	TAC-Online
IAC-Default	TAC-Default

Каждый из этих объектов имеет тот же формат, что и TVR. IAC и TAC не являются обязательными с точки зрения спецификаций EMV. Но ведущие платежные системы требуют их присутствия на карте и в терминале.

TAC загружаются в терминал обслуживающим банком (например, Visa и MasterCard определяют обязательные TAC для обслуживающих банков). Эти объекты зависят от типа терминала и от карточного продукта. IAC

определяются эмитентом карты и заносятся на карту во время её персонализации. Эти объекты определяют политику эмитента в области обеспечения безопасности своих операций. Назначение объектов поясняется в нижеприведенной таблице (в этой таблице используется синоним словосочетания «обслуживающий банк» –эквайер).

Объект	Источник	Назначение	По умолчанию
IAC-Denial	Эмитент	Условия, при которых транзакция должна быть отвергнута	Все нули
IAC-Online	Эмитент	Условия, при которых транзакция должна быть отправлена на авторизацию эмитенту	Все единицы
IAC-Default	Эмитент	Условия, при которых терминал, неспособный обслужить транзакцию в online, должен её отвергнуть	Все единицы
TAC- Denial	Эквайер	Условия, при которых транзакция должна быть отвергнута	Все нули
TAC- Online	Эквайер	Условия, при которых транзакция должна быть отправлена на авторизацию эмитенту	Все нули
TAC-Default	Эквайер	Условия, при которых терминал, неспособный обслужить транзакцию в online, должен её отвергнуть	Все нули

Хотя по умолчанию все биты объектов TAC-Online и TAC-Default равны 0, в то же время EMVCo настоятельно рекомендует, чтобы обслуживающий банк определил по крайней мере следующие признаки:

- не выполнена офлайновая аутентификация данных карты
- офлайновая аутентификация данных карты провалилась

Терминал формирует свое решение о способе обработки транзакции, сравнивая биты, установленные в TVR, IAC и TAC, следующим образом.

1. Если в TVR есть единичные биты и соответствующие им биты в IAC-Denial и TAC-Denial также установлены в 1[1], то транзакция должна быть отвергнута без попытки выполнения онлайновой авторизации. В противном случае, терминал переходит к шагу 2, если терминал способен выполнить транзакцию в online, или к шагу 3, когда терминал работает только в offline.

[1] Т.е. результат операции побитового логического умножения TVR и результата побитового логического сложения IAC и TAC не равен 0.

2. Когда в TVR есть единичные биты и соответствующие им биты в IAC-Online и TAC-Online также установлены в 1, терминал считает, что транзакцию нужно отправить на авторизацию эмитенту. В противном случае, терминал предлагает карте одобрить транзакцию в офлайновом режиме.

3. Если в TVR есть единичные биты и соответствующие им биты в IAC-Default и TAC-Default также установлены в 1, то терминал считает, что транзакция должна быть отклонена. В противном случае, терминал предлагает карте одобрить транзакцию в офлайновом режиме.

Принятое решение о способе обработки транзакции терминал сообщает карте в первой команде GENERATE AC. В первой команде GENERATE AC терминал может запросить от карты формирование одной из следующих криптограмм.

1. Криптограммы AAC (Application Authentication Cryptogram), если транзакция должна быть отвергнута без попытки выполнения онлайновой авторизации.

2. Криптограммы ARQC (Authorization Request Cryptogram), когда терминал принял решение отправить транзакцию на авторизацию эмитенту.

3. Криптограммы TC (Transaction Certificate), если терминал предлагает карте одобрить транзакцию в офлайновом режиме

Карта получает от терминала в команде GENERATE AC не только тип транзакции, запрашиваемой терминалом, но и данные, которые необходимы карте для выполнения собственных процедур управления рисками. Терминал узнает о том, какие данные требуются карте для выполнения процедур управления рисками и вычисления криптограммы через списки объектов данных, которые описаны в следующем разделе.

Управление рисками карты

Важная роль в процессе обработки транзакции отводится карте, которой эмитентом делегируются функции, связанные с принятием решения о способе завершения транзакции. Карта, как и терминал, выполняет собственные процедуры управления рисками (Card Risk Management – CRM). На основе выполненных проверок карта проводит анализ полученных результатов и выносит своё решение (точнее, решение эмитента) о способе завершения транзакции.

По аналогии с терминалом результаты своих проверок карта записывает в элемент данных, который называется Card Verification Results (CVR). Этот

элемент данных используется только картой (и эмитентом[1]) для принятия решений по результатам обработки транзакций. Важное отличие CVR от TVR состоит в том, что в CVR часто хранится не только информация о текущем состоянии, но и часть истории, связанной с использованием карты, которая может потребоваться эмитенту.

Выполняемые картой процедуры управления рисками можно разделить по следующим типам:

- определение статуса проверки PIN-кода в offline
- анализ результатов выполнения предыдущей транзакции
- проверка офлайновых лимитов, ограничивающих количество последовательно выполненных картой офлайновых транзакций
- проверка офлайновых лимитов, ограничивающих объем средств, потраченных в последовательно выполненных картой офлайновых транзакциях
- проверка статуса и результата выполнения команд скрипт-процессинга
- проверка специальных условий обслуживания карты (признаков, которые сигнализируют о необходимости выполнения текущей транзакции в online, определение факта того, что карта ещё ни разу не была авторизована эмитентом и т. п.)

По мере выполнения процедур управления рисками карта формирует CVR. После завершения всех процедур карта выполняет анализ всех ситуаций, зафиксированных в CVR. Целью такого анализа является выработка картой решения об обработке транзакции. Решение формируется на основе правил, определенных эмитентом карты. Для этого в платежном приложении определено множество объектов данных, которые называются Card Issuer Action Codes (CIAC) и состоит из трех объектов с суффиксами Denial, Online и Default, т. е. CIAC-Denial, CIAC-Online и CIAC-Default.[2]

Каждый из объектов CIAC имеет формат, похожий на формат CVR. Как обсуждалось ранее, в CVR хранится некоторая информация о текущем состоянии, которая не используется в выработке решений картой. В связи с этим в CIAC определены только биты CVR, информирующие о результатах

[1] CVR передается эмитенту в качестве компонента объекта Issuer Application Data.

[2] Не все платежные приложения поддерживают логику Card Risk Management с использованием CVR и CIAC. Например, приложение Visa использует другой алгоритм, базирующийся на указании эмитентом признаков, определяющих решение карты, в специальных объектах, записываемых в платежное приложение во время его персонализации. Но для очень большого числа платежных приложений применяется логика CVR и CIAC.

выполнения предыдущих транзакций и исключительных ситуациях, зафиксированных для текущей транзакции.[1]

Выработка картой решения об обработке транзакции включает следующие шаги.

1. Если терминал запрашивает у карты криптограмму AAC, то карта не анализирует CVR и генерирует запрашиваемую криптограмму.

2. Если терминал запрашивает у карты криптограмму ARQC, то действия карты зависят от типа терминала. Когда терминал способен выполнить транзакцию в online, терминал соглашается с решением терминала и формирует криптограмму ARQC. В случае офлайнового терминала транзакция отвергается и формируется криптограмма AAC.

3. Когда терминал запрашивает у карты криптограмму TC, и в CVR установлены биты, которым найдено соответствие в CIAC-Denial[2], транзакция отвергается, карта формирует криптограмму AAC и возвращает её терминалу. В противном случае карта переходит либо к шагу 4, если терминал способен выполнить транзакцию в online, либо к шагу 5 – в противном случае.

4. Карта проверяет, не соответствуют ли ненулевым битам в CVR биты, установленные в CIAC-Online. Если такое соответствие найдено, то карта считает, что транзакция нужно отправить на авторизацию эмитенту, и формирует криптограмму ARQC. В противном случае карта считает, что транзакцию нужно одобрить в офлайновом режиме, и формирует криптограмму TC.

5. Если в CVR установлены биты, которым найдено соответствие в CIAC-Default, транзакция отвергается, и карта формирует криптограмму AAC. В противном случае карта предлагает одобрить транзакцию в офлайновом режиме, и формирует криптограмму TC.

Рассмотрим, каким образом эмитент может управлять решением карты об обработке транзакции с помощью признаков, установленных в CIAC, на примере офлайновых счетчиков. В большинстве платежных приложений существует два вида счетчиков:

- однобайтный счетчик количества последовательных операций, выполненных в offline (Consecutive Offline Transaction Number – COTN)

[1] Опять же, нет правил без исключений. Результат выполнения предыдущих транзакций может не учитываться (зависит от приложения). Но для некоторых платежных приложений (например, созданных в соответствии со спецификациями EMV CCD) – это обязательное условие.

[2] Т. е. результат операции побитового логического умножения CVR и CIAC не равен 0.

- сумма всех последовательных платежных операций, выполненных в offline (Consecutive Offline Transaction Amount – COTA)

Каждому из счетчиков соответствует два лимита, определяемых эмитентом: нижний лимит и верхний лимит. Карта устанавливает в CVR признаки превышения заданных лимитов. Любой из счетчиков может использоваться для ограничения объема средств, потраченных в последовательно выполненных картой офлайновых транзакциях. Например, эмитент хочет использовать для этой цели количество последовательных операций, выполненных в offline, как это показано на рис. 4.

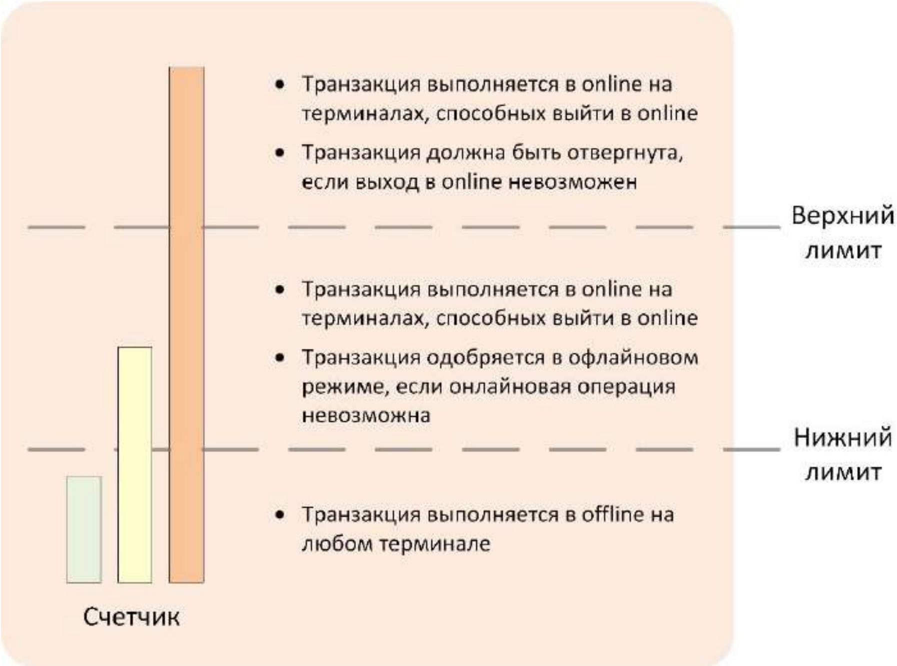


Рис. 4. Типичное использование офлайнового счетчика.

Логику ограничения последовательно выполненных картой офлайновых транзакций можно описать следующим образом:

- если счетчик меньше или равен нижнему лимиту, то транзакция должна быть выполнена в offline

- когда счетчик больше нижнего лимита, но не выше верхнего лимита, транзакция должна аутентифицирована эмитентом на терминалах, которые способны выйти в online, или должна быть одобрена в офлайновом режиме, если онлайновая операция невозможна[1]

[1] Потому что терминал работает только в offline, или из-за того, что терминал не может в данный момент выполнить транзакцию в online (unable to go online).

- если счетчик превышает верхний лимит, то транзакция должна быть выполнена в online на терминалах, которые способны выйти в online, или должна быть отвергнута, если онлайновая операция невозможна

Чтобы карта могла поддержать такую логику ограничения последовательно выполненных офлайновых транзакций, эмитент должен установить в CIAC-Online биты «Превышен нижний предел офлайновых транзакций» и «Превышен верхний предел офлайновых транзакций», а в CIAC-Default –бит «Превышен верхний предел офлайновых транзакций».

Тогда, если терминал запрашивает криптограмму TC, карта увеличивает значение счетчика на 1, сверяет его лимитами и устанавливает в CVR признак «Превышен нижний предел офлайновых транзакций», если счетчик больше нижнего предела, и признак «Превышен верхний предел офлайновых транзакций», если счетчик больше верхнего лимита. Если в CVR не установлен ни один из признаков превышения лимита, то карта сформирует криптограмму TC независимо от типа терминала. В случае установки в CVR признака превышения нижнего лимита (но не верхнего) для онлайнового терминала будет найдено соответствие признаков в CVR и CIAC-Online, и карта вернет криптограмму ARQC, а для офлайнового терминала – криптограмму TC, поскольку в CIAC-Default признак «Превышен нижний предел офлайновых транзакций» не установлен. Наконец, если в CVR установлены оба признака превышения лимитов, то карта сформирует криптограмму ARQC для онлайнового терминала и криптограмму AAC для офлайнового терминала, поскольку в CIAC-Default установлен признак «Превышен верхний предел офлайновых транзакций».

Онлайновая обработка транзакции

Действия, выполняемые обслуживающим банком или эмитентом, не должны описываться в этом документе. Хотя бы потому, что эмулятор терминала ECV не предназначен для связи с обслуживающим банком и эмитентом. Однако, по многим причинам, которые будут объяснены позже, следует пояснить логику обработки транзакции в онлайновом режиме. Следует иметь в виду, что далее приведено самое общее описание, которое может быть не применимо к ряду платежных приложений.

Если карта в ответе на первую команду GENERATE AC указывает, что транзакцию нужно отправить на авторизацию эмитенту, терминал передает хосту обслуживающего банка сообщение, которое содержит информацию, относящуюся к карте и результатам обработки транзакции. На основе этих данных хост обслуживающего банка формирует авторизационный запрос эмитенту (сообщение x100 стандарта ISO 8583), содержащий такие данные:

- криптограмма ARQC
- информация о карте (PAN, срок действия карты и т. п.)

- информация о терминале (идентификатор, тип и т. п.)
- последовательный номер транзакции ATC
- элемент Issuer Authentication Data, сформированный картой

При получении авторизационного запроса эмитент проверяет, является ли карта, по которой проводится операция, подлинной. Для этого эмитент вычисляет сессионный ключ генерации криптограммы, после чего использует его и данные транзакции для получения криптограммы. Полученная криптограмма сравнивается с предоставленным значением ARQC. Если значения совпадают, аутентификация карты считается завершенной успешно, а сама карта – подлинной.

Далее эмитент проводит стандартные проверки (проверка активности карты, истории использования карты, отсутствия карты в списке заблокированных карт, наличие средств на карт-счете, и т. п.).

Данные, получаемые эмитентом в авторизационном запросе, позволяют ему принять правильное решение по способу завершения текущей операции, повлиять на выполнение следующей транзакции и произвести коррекцию данных карты с помощью команд скрипт-процессинга и (или) элемента данных Card Status Update (CSU).

После принятия решения о результате операции эмитент генерирует элемент данных Issuer Authentication Data. Этот элемент содержит криптограмму эмитента ARPC, которая используются картой для аутентификации эмитента, и Card Status Update (CSU), который указывает, одобрена или отклонена транзакция эмитентом, а также определяет действия, которые с точки зрения эмитента должны быть выполнены картой. В авторизационный ответ обслуживающему банку (сообщение x100 стандарта ISO 8583) помещается элемент данных Issuer Authentication Data, код авторизации эмитента, а также команды скрипт-процессинга, если они должны быть переданы карте.[1]

Если терминал не получил авторизационный ответ или получил его слишком поздно, терминал продолжает обрабатывать транзакцию, считая, что запрос невозможно передать эмитенту (эта ситуация обычно называется «unable to go online»). В любом случае, терминал должен получить криптограмму приложения от карты с помощью второй команды GENERATE AC. Однако, перед выполнением этой команды терминал должен передать карте команды критичного скрипт-процессинга, если они присутствуют в авторизационном ответе.

Во второй команде GENERATE AC терминал передает карте данные, полученные от эмитента (код авторизации эмитента, элемент Issuer

[1] Элементы данных CSU и Issuer Authentication Data не являются обязательными для платежного приложения. Для некоторых приложений обязательна только криптограмма эмитента.

Authentication Data)[1] и уточненные данные своих проверок (TVR). Во второй команде GENERATE AC терминал может запросить от карты формирование одной из следующих криптограмм.

1. Криптограммы AAC, если транзакция должна быть отклонена.

2. Криптограммы TC, если терминал считает, что транзакцию следует одобрить.

Процесс принятия картой решения после получения второй команды GENERATE AC включает следующие шаги.

1. Если терминал запрашивает криптограмму AAC, то карта генерирует запрашиваемую криптограмму.

2. Когда терминал информирует о ситуации «unable to go online», карта проверяет, не соответствуют ли ненулевым битам в CVR биты, установленные в CIAC-Default. Если такое соответствие найдено, то транзакция отвергается, и карта формирует криптограмму AAC. В противном случае карта может одобрить транзакцию после проведения дополнительных проверок.

3. Карта аутентифицирует эмитента (проверяет значение криптограммы ARPC в элементе Issuer Authentication Data). Если аутентификация эмитента не выполнена, то обычно транзакция отвергается (формируется криптограмма AAC).

4. Если аутентификация эмитента выполнена успешно, дальнейшие действия карты определяются выбором эмитента, который определен в Card Status Update (CSU). Карта формирует криптограмму TC или AAC в зависимости от решения эмитента об обработке транзакции, и выполняет действия, которые определил эмитент в CSU (например, установку счетчика оставшихся предъявлений PIN-кола, сброс офлайновых счётчиков и т. п.). Кроме этого, карта сбрасывает признаки особых ситуаций, которые были зафиксированы при выполнении предыдущей или текущей транзакции.

Решение карты, принятое при выполнении второй команды GENERATE AC, является окончательным.

После завершения второй команды GENERATE AC терминал должен передать карте команды некритичного скрипт-процессинга, если они присутствуют в авторизационном ответе.

[1] Если онлайновая обработка невозможна (ситуация «unable to go online»), то терминал сообщает об этом через код авторизации эмитента (элемент Issuer Authentication Data при этом обнулен).

Транзакция в бесконтактном режиме

Интерес к бесконтактным картам можно объяснить несколькими их техническими преимуществами:

- удобство использования карты (карту не нужно передавать кассиру, правильно ориентировать и вставлять в прорезь ридера, даже можно не вытаскивать из бумажника[1])
- более высокая скорость выполнения транзакции
- более высокая надежность использования карт и терминалов – из-за отсутствия механического контакта карты и терминала обеспечивается более низкий уровень их физического износа
- лучшая защищенность бесконтактных терминалов от случаев вандализма

Компания EMVCo уже давно прорабатывает вопрос о создании единого бесконтактного приложения EMV Contactless Application – аналога приложения Common Payment Application (CPA) для контактных карт. Однако компания EMVCo не торопится с разработкой этого стандарта. Причина, озвучиваемая EMVCo, – недостаток опыта использования бесконтактных карт. Каждый год проводится внутреннее обсуждение вопроса о целесообразности разработки стандарта, но положительного решения до сих пор нет. Появление стандарта на единое бесконтактное приложение в будущем не совсем очевидно. Поэтому EMVCo, предваряя появление такого стандарта, разработала спецификацию EMV Contactless Specifications for Payment Systems – Entry Point Specification (в дальнейшем будем для краткости называть его Entry Point Specification), решающую ряд вопросов.

'Этот стандарт определяет общую схему обработки транзакции по бесконтактной карте на стороне терминала (Entry Point) и достаточно подробно останавливается на описании процедур, предшествующих выбору приложения. На рис. 5 показана общая схема обработки транзакции в бесконтактном режиме. Не останавливаясь на деталях обработки отметим, что после предварительной обработки и активации протокола терминал начинает процедуру выбора бесконтактного приложения. Если терминал поддерживает единственное бесконтактное приложение, то он сразу выбирает это приложение с помощью команды SELECT. Иначе, терминал выбирает приложение PPSE (Proximity Payment System Environment), которое имеет идентификатор 2PAY.SYS.DDF01, и получает в ответ объект FCI, который содержит информацию о бесконтактных приложениях на карте. Далее

---

[1] Хотя идея о том, что карту можно не вытаскивать из бумажника, до сих переходит из книги в книгу, из статьи в статью, она трудноосуществима. Во-первых, хороший бумажник значительно снижает возможность распознавания карты терминалом из-за снижения мощности сигнала. Во-вторых, если в бумажнике есть другие бесконтактные карты, терминал скорее всего будет неспособен распознать карту из-за возникающих коллизий.

терминал определяет, какие из поддерживаемых им приложений находятся на карте, локализует приложение с наивысшим приоритетом и второй командой SELECT выбирает его.

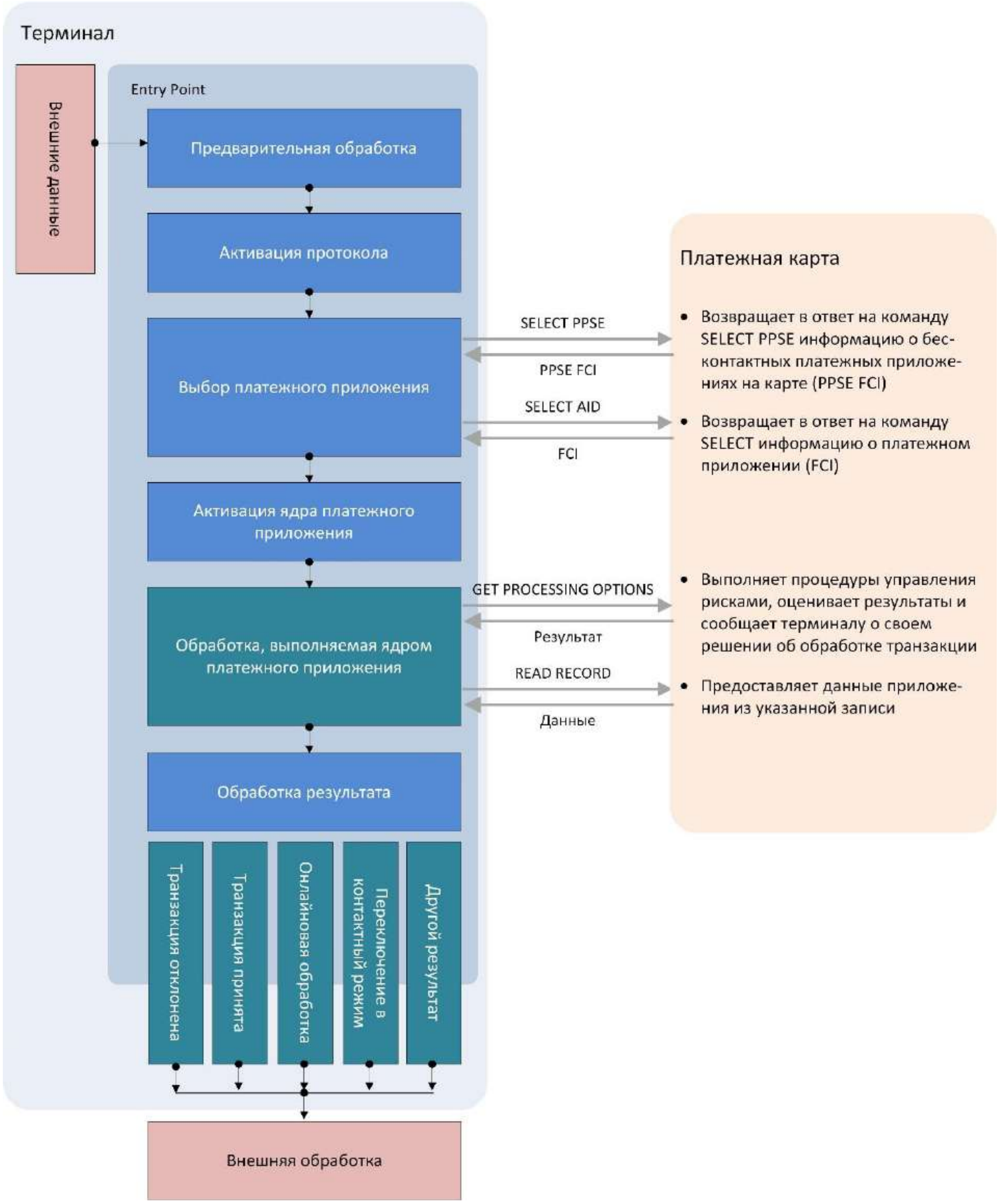


Рис. 5. Обработка транзакции в бесконтактном режиме.

После выбора приложения происходит активация ядра, соответствующего приложению, которому Entry Point терминала передает управление. Ядро завершает обработку формированием результата для Entry Point. Возможные результаты обработки ядра – отклонение транзакции или одобрение в офлайновом режиме, направление транзакции для авторизации эмитенту, требование переключения в контактный режим и т. п.

Одна из основных особенностей работы в бесконтактном режиме состоит в том, что обычно транзакция выполняется за одно прикосновение к терминалу.[1] Кроме того, требуется, чтобы время нахождения карты в зоне считывания терминала было минимальным. Например, MasterCard настаивает на том, чтобы карта и терминал успели обменяться данными за 150 мсек (хотя сам эти требования и нарушает). В результате реализация бесконтактной транзакции в платежном приложении характеризуется следующими особенностями:

- для верификации владельца карты не может использоваться метод офлайновой проверки PIN-кода[2]
- процедуры управления рисками, выполняемые терминалом, учитывают, что транзакция выполняется в бесконтактном режиме
- в случае онлайновой авторизации транзакции ответ эмитента не проверяется картой, команды скрипт-процессинга не выполняются (карта уже недоступна для терминала), а также отсутствует возможность выполнения действий, определяемых эмитентом в CSU (например, установки счетчика предъявлений PIN-кола, сброса офлайновых счётчиков и т. п.)
- набор команд, используемых терминалом (ядром обработки приложения) для выполнения транзакции, обычно минимизируется, чтобы уменьшить время нахождения карты в зоне считывания терминала[3]

---

[1] Ряд платежных систем (например, AmEx) пошел по пути реализации транзакции за два прикосновения к терминалу. Второе прикосновение нужно после поступления ответа эмитента для его обработки. В этом случае бесконтактная обработка практически не отличается от контактной. Такой метод не только неудобен для владельца карты, но и замедляет оплату товаров (услуг). Поэтому большинство платежных систем всё же используют единственное прикосновение. Этот метод и описывается далее.

[2] Офлайновая проверки PIN-кода нежелательна по многим причинам. Во-первых, это небезопасно. Во-вторых, неудобно для владельца карты. Недаром, общая черта спецификаций бесконтактных приложений – отказ от предъявления PIN-кода в offline. Бесконтактные приложения используют два метода верификации – предъявление PIN-кода в online (при направлении транзакции для авторизации эмитенту) и подпись (в офлайновом режиме).

[3] Например, в ряде платежных приложений не используется команда GENERATE AC, а данные, необходимые для одобрения транзакции эмитентом (криптограмма и другие параметры транзакции), предоставляются терминалу в команде GET PROCESSING OPTIONS.

Верификация владельца бесконтактной карты ограничивается несколькими методами:

- проверка зашифрованного PIN-кода, выполняемая эмитентом (онлайновый PIN-код)
- получение подписи владельца карты
- верификация владельца карты не требуется (актуально в ряде случаев, когда сумма транзакции мала или необходимо обеспечить высокую скорость платежей)

Кроме того, для мобильного устройства существует особый случай верификации. Мобильное устройство может сообщить терминалу, что верификация владельца карты уже выполнена (на устройстве введен PIN-код или предъявлен отпечаток пальца).

Необходимо отметить, что для выбора метода верификации бесконтактная карта может не использовать CVM List, определяющий список методов и условий верификации владельца карты для контактного режима. Для целого ряда приложений используются другие объекты, которые определяют выбор метода верификации для бесконтактного режима.

Процедуры управления рисками, выполняемые терминалом в бесконтактном режиме, достаточно просты. Во-первых, терминал должен сравнить сумму транзакции с пороговой суммой Contactless Transaction Limit. Если сумма транзакции превышает это значение, то бесконтактная транзакция не выполняется. Во-вторых, терминал сравнивает сумму транзакции с пороговой суммой CVM Required Limit. Когда сумма транзакции превышает это пороговое значение, терминал считает, что должна быть выполнена верификация владельца карты.

Обычно, после этого терминал выдает команду GET PROCESSING OPTIONS, с которой передаются данные, необходимые карте для принятия решения о способе обработки транзакции. Эти данные определяются списком PDOL и определяют не только параметры транзакции, но и возможности терминала, а также результат предварительной обработки терминалом транзакции в бесконтактном режиме.

Карта выполняет процедуры управления рисками и формирует CVR. При установке отдельных битов CVR учитываются признаки особых ситуаций, возникших при обработке предыдущей транзакции в контактном режиме (признаки особых ситуаций в бесконтактном режиме обычно не устанавливаются).

После завершения всех процедур карта принимает решение относительно завершения транзакции в соответствии с массивом CIAC[1]. Может быть принято одно из следующих решений:

- транзакция одобряется в офлайновом режиме
- требуется онлайновая авторизация транзакции эмитентом
- транзакция должна быть отклонена
- требуется переключение в контактный режим

Последнее решение карта может принять в том случае, когда из-за целого ряда факторов выполнение транзакции в бесконтактном режиме невозможно (или нежелательно). Но решение о переключении в контактный режим принимается только в том случае, если ли терминал поддерживает контактный режим (в противном случае транзакция отклоняется). Таким образом, решение платежного приложения всегда согласовано с возможностями терминала и является окончательным.

В зависимости от принятого решения карта может предоставить разные данные. Единственный элемент, который всегда входит в состав возвращаемых данных, информирует терминал о выборе карты. Если карта возвращает криптограмму транзакции, то предоставляются также и другие данные, которые могут потребоваться терминалу для продолжения обработки (например, для формирования авторизационного запроса эмитенту). Среди этих данных присутствует элемент Application File Locator (AFL), содержащий ссылки на данные, которые терминал должен прочитать, чтобы успешно выполнить транзакцию.

После чтения данных, определяемых AFL, с помощью команды READ RECORD обработка ядра завершается формированием результата. Начиная с этого момента времени, карта уже не нужна для дальнейшей обработки и может быть удалена из зоны считывания терминала.

Дальнейшие действия терминала зависят от результата обработки ядра. Возможны следующие варианты.

1. Требуется переключение в контактный режим обработки. Терминал предпринимает попытку выполнить транзакцию в контактном режиме.

2. Транзакция отклонена. Терминал завершает обработку транзакции.

[1] Следует иметь в виду, что CIAC для бесконтактного режима могут отличаться от CIAC, применяемых для обработки транзакции в контактном режиме. Или CIAC вообще не используются для бесконтактного режима.

3. Должна быть выполнена онлайновая обработка. Терминал запрашивает ввод PIN-кода, если это требуется, формирует авторизационный запрос для эмитента и пересылает этот запрос хосту. Основное отличие онлайновой обработки транзакции в бесконтактном режиме заключается в том, что ответ эмитента проверяет не карта (она уже удалена), а терминал. Конечно, терминал не может выполнить аутентификацию эмитента. Его решение об обработке транзакции должно соответствовать выбору эмитента.

Глава 2

Возможности

Рабочее место комплекса тестирования ECV – это специальное устройство чтения смарт-карт с установленной лицензионной картой и программа проверки платежной карты, которая может функционировать только в том случае, если обнаружит подключенное специальное устройство чтения смарт-карт. К рабочему месту комплекса тестирования могут быть подключены и другие устройства чтения смарт-карт, но наличие специального устройства с установленной лицензионной картой обязательно. Это связано только с тем, что компания СканТек лицензирует использование комплекса тестирования ECV с помощью лицензионной карты.

Все устройства чтения смарт-карт, с которыми работает комплекс тестирования ECV – это PCSC-устройства. PCSC (правильно PC/SC, но в России уже давно принято наименование PCSC без слэша) – это сокращенное наименование (Personal Computer/ Smart Card) спецификации Microsoft для интеграции смарт-карт в среду персональных компьютеров. Microsoft имплементировал PCSC в Windows 200x/XP (и даже сделал PCSC доступным в Windows NT/9x!). Интересно, что свободная реализация (free software) существует даже для Linux (а также других Unix) в виде PC/SC Lite, а не совсем легальная версия PC/SC Lite существует даже для Mac OS X.

Все эти рассуждения интересны, но программа проверки платежной карты функционирует только под управлением операционной системы Windows. Рекомендуется версия Windows 10, но в любом случае версия операционной системы должна быть не ниже Windows 2000/XP. Любые версии программы под управлением Linux (и других Unix), а также Mac OS, не планируются, и вряд ли будут когда-нибудь осуществлены.

После запуска программа проверки платежной карты на экране дисплея персонального компьютера, функционирующего под управлением Windows, появляется окно программы, которое выглядит так, как показано на рис. 6. Конечно, окно выглядит не совсем так (и даже совсем не так). Потому что на рисунке выделены группы управляющих элементов, с помощью которых осуществляется регулирование процессом исследования EMV-приложения и наблюдение за ходом его выполнения.

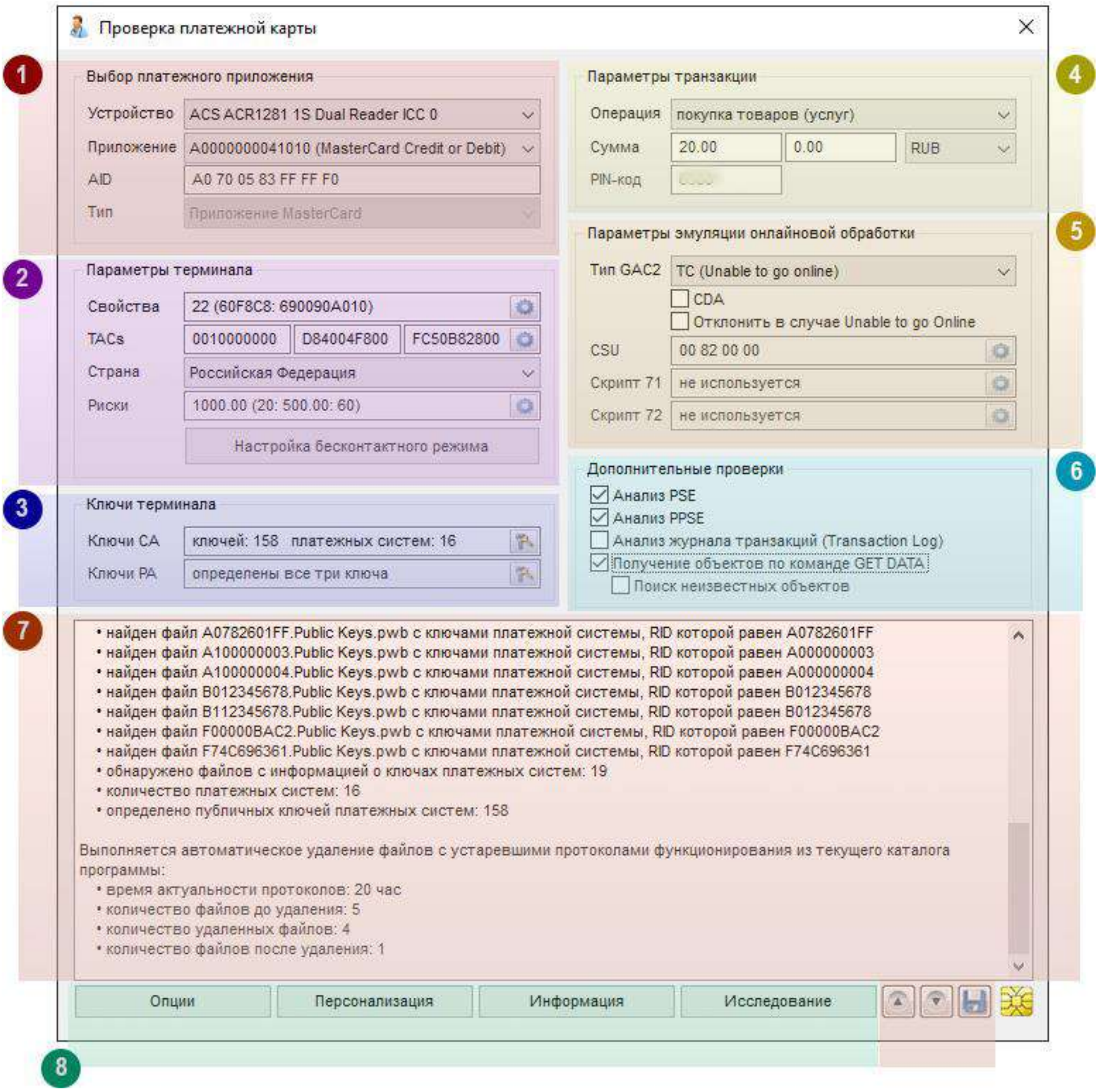


Рис. 6. Главное окно программы проверки платежной карты.

На рисунке выделены следующие группы управляющих элементов, объединённых по функциональным признакам.

1. Выбор платежного приложения, исследование которого должно быть выполнено.

2. Определение параметров и возможностей терминала по обработке платежной транзакции.

3. Определение ключей терминала, необходимых для обработки транзакции, а также ключей, которые требуются эмулятору терминала для моделирования онлайновой обработки.

4. Основные параметры платежной транзакции и, возможно, персональная идентификация владельца карты.

5. Параметры и особенности эмуляции онлайновой обработки.

6. Дополнительные проверки, которые должны быть выполнены для карты или платежного приложения.

7. Журнал событий (протокол), происходящих в процессе функционирования комплекса тестирования, а также кнопки для управления журналом.

8. Кнопки управления комплексом тестирования.

Далее подробно рассматриваются возможности программы в соответствии с выделенными группами управляющих элементов. Это необходимо для того, чтобы пользователь смог понять, какие возможности доступны при анализе EMV-приложения. Хотя программа проверки платежной карты и обстоятельно «документирована» (при наведении курсора мыши на любой управляющий элемент отображается подсказка, которая служит дополнительным средством обучения пользователя), но это не означает, что все возможности будут прозрачны для пользователя и он поймет логику разработчика.

Выбор платежного приложения

Одна из основных проблем, возникающих при анализе карты, состоит в том, чтобы определить, какие платежные приложения есть на карте. Как уже говорилось ранее, для POS-терминала существует два метода выбора приложения: прямой выбор и выбор из списка (PSE или PPSE). Эмулятор терминала позволяет сделать то же самое, но с некоторыми дополнительными возможностями. Определение параметров для выбора анализируемого платежного приложения иллюстрируется с помощью рис. 7.

Но сначала нужно выбрать PCSC-устройство, в которое будет установлена исследуемая карта. Для этого служит combo-box[1], который содержит список всех устройств чтения смарт-карт, подключённых к компьютеру. Список устройств создается при запуске программы проверки платежной карты. Если новое PCSC-устройство подключается динамически в процессе функционирования, то достаточно построить список устройств заново.

[1] Combo-box – это элемент управления, представляющий собой комбинацию списка элементов и строки редактирования. В дальнейшем такой элемент управления будет называться combo-box, потому что для IT-специалистов перевод может привести к неправильному толкованию термина.

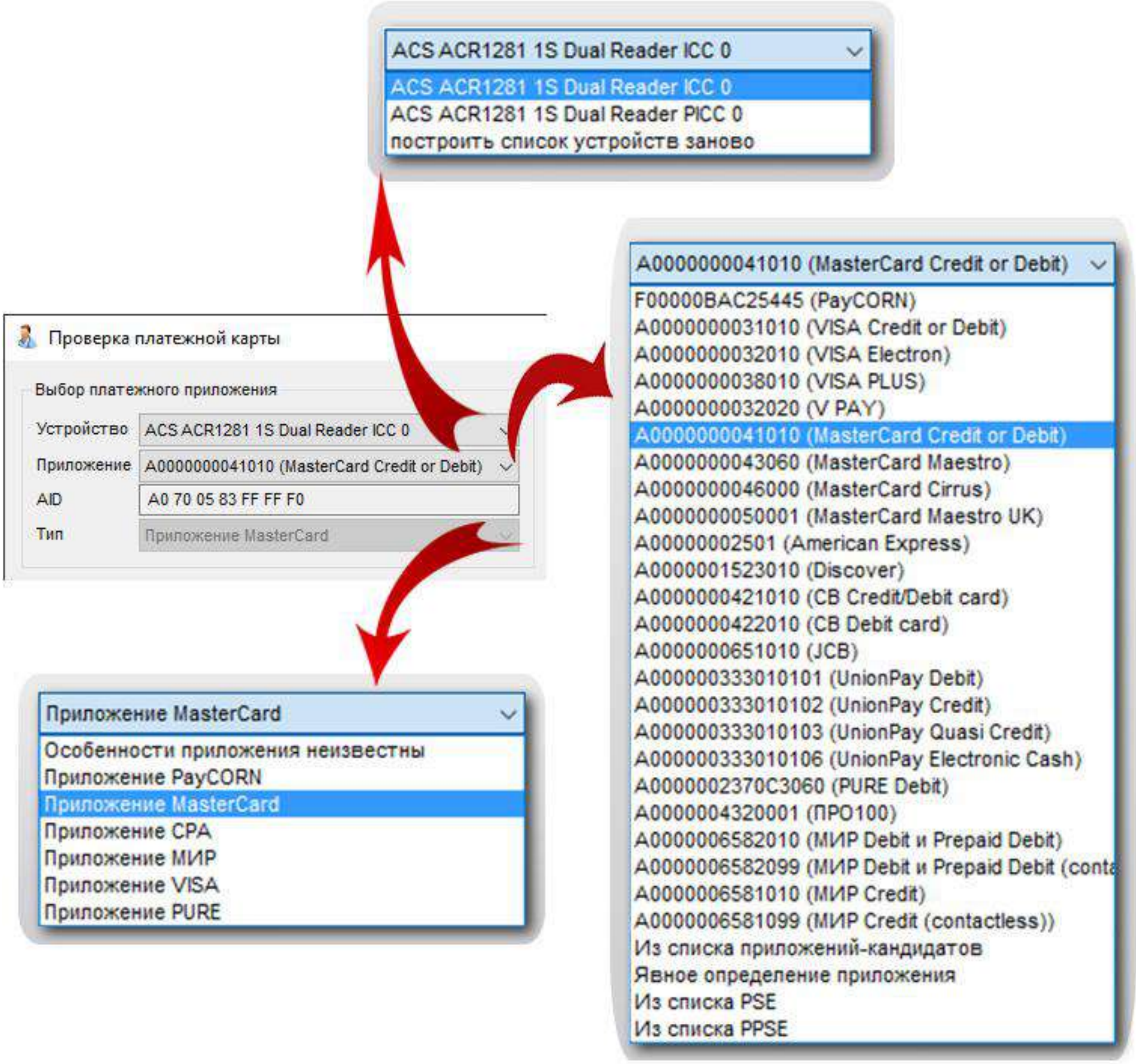


Рис. 7. Выбор устройства и платежного приложения.

Применяемый метод выбора анализируемого платежного приложения указывается параметрами, которые определяются с помощью оставшихся управляющих элементов.

Во-первых, можно явно указать, что на карте должно быть выбрано приложение с заданным идентификатором (AID). Для этого в combo-box «Приложение» выбирается строка с одним из стандартных AID, соответствующим платежному приложению.

Во-вторых, можно указать, что должен быть построен список приложений-кандидатов и анализируемое приложение должно быть выбрано из этого списка. Список приложений- кандидатов строится следующим образом:

- делается попытка отыскать на карте все приложения, AID которых начинается с RID[1] платежных систем, перечисленных в списке combo-box
- любое найденное приложение проверяется и устанавливается, является ли оно платежным[2]
- если приложение является платежным, то оно заносится в список приложений- кандидатов

После завершения процедуры построения списка приложений- кандидатов пользователю предлагается выбрать исследуемое приложение из списка, как это показано на рис. 8.

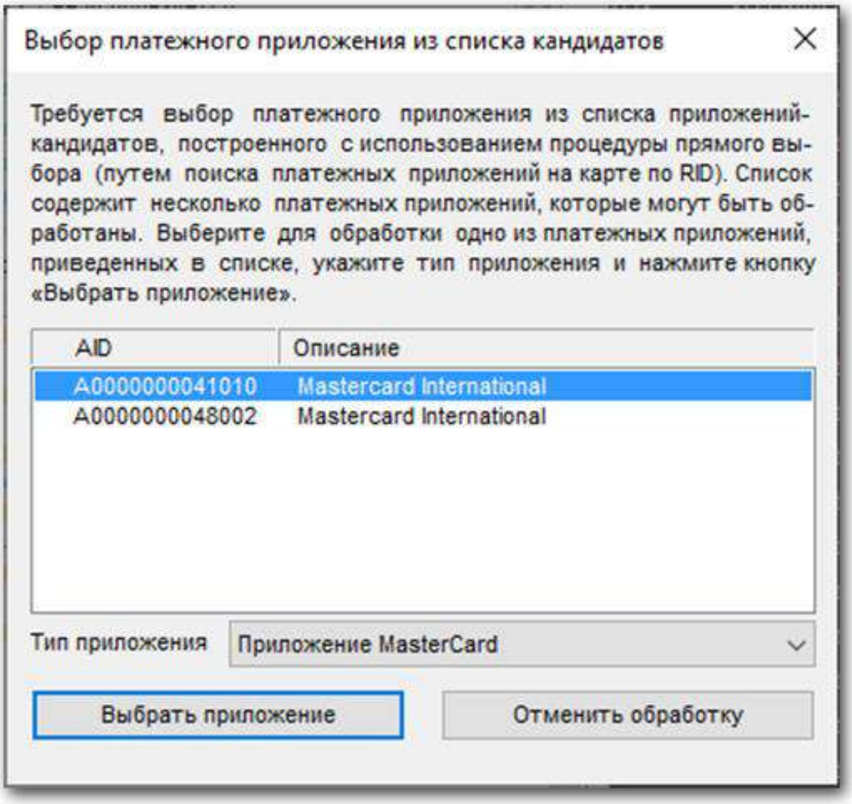


Рис. 8. Выбор приложения из списка приложений-кандидатов.

[1] RID – это первые пять байт AID, которые назначаются платежной системе и однозначно её идентифицируют. Например, для MasterCard назначен RID в виде A000000004 и AID всех карточных продуктов MasterCard должны начинаться с этого RID.

[2] Не все приложения на карте с RID платежной системы обязательно являются платежными. На карте могут существовать специальные сервисные приложения, которые относятся к платежной системе, но не могут использоваться терминалом для проведения транзакции.

В-третьих, интересующее приложение может быть определено явно. Для этого нужно в списке combo-box выбрать строку «Явное определение приложения», после чего в элементе редактирования «AID» ввести AID приложения.[1] В этом случае может потребоваться выбрать тип приложения из списка combo-box «Тип»

Наконец, приложение может быть выбрано из списка PSE (для контактного режима) или списка PPSE (для бесконтактного режима), если они есть на карте. В этом случае окончательный выбор приложения остается за пользователем. Он должен выбрать анализируемое приложение из списка, который будет отображен на экране (см. рис. 8).

Определение параметров терминала

Для выполнения транзакции эмулятор терминала должен быть настроен соответствующим образом. Дело в том, что выполнение транзакции зависит от типа и возможностей терминала, политики эквайера в области безопасности платежных операций, и еще многих других параметров, которые в реальный терминал обычно загружает обслуживающий банк. Для эмулятора терминала все эти параметры могут меняться динамически перед выполнением транзакции, что позволяет проверять различные варианты поведения платежных приложений.

Определение параметров терминала демонстрируется с помощью рис. 9. Пользователь может определить следующее.

- Тип терминала, его возможности, а также дополнительные возможности, которые определяют физические особенности POS-терминала и типы разрешенных транзакций.
- Условия принятия решения об обработке транзакции, которые устанавливаются для терминала в виде Terminal Action Codes (TACs).
- Страну, в которой расположен терминал (платежное приложение может быть настроено на принятие решения об обработке транзакции в зависимости от того, является ли транзакция внутренней или международной).
- Параметры управления рисками терминала для контактных транзакций.
- Параметры управления рисками для бесконтактных транзакций, которые обычно зависят от платежной системы.

[1] Элемент редактирования «AID» используется только при явном определении приложения. Во всех остальных случаях он недоступен для ввода и на его содержимое можно не обращать внимания.

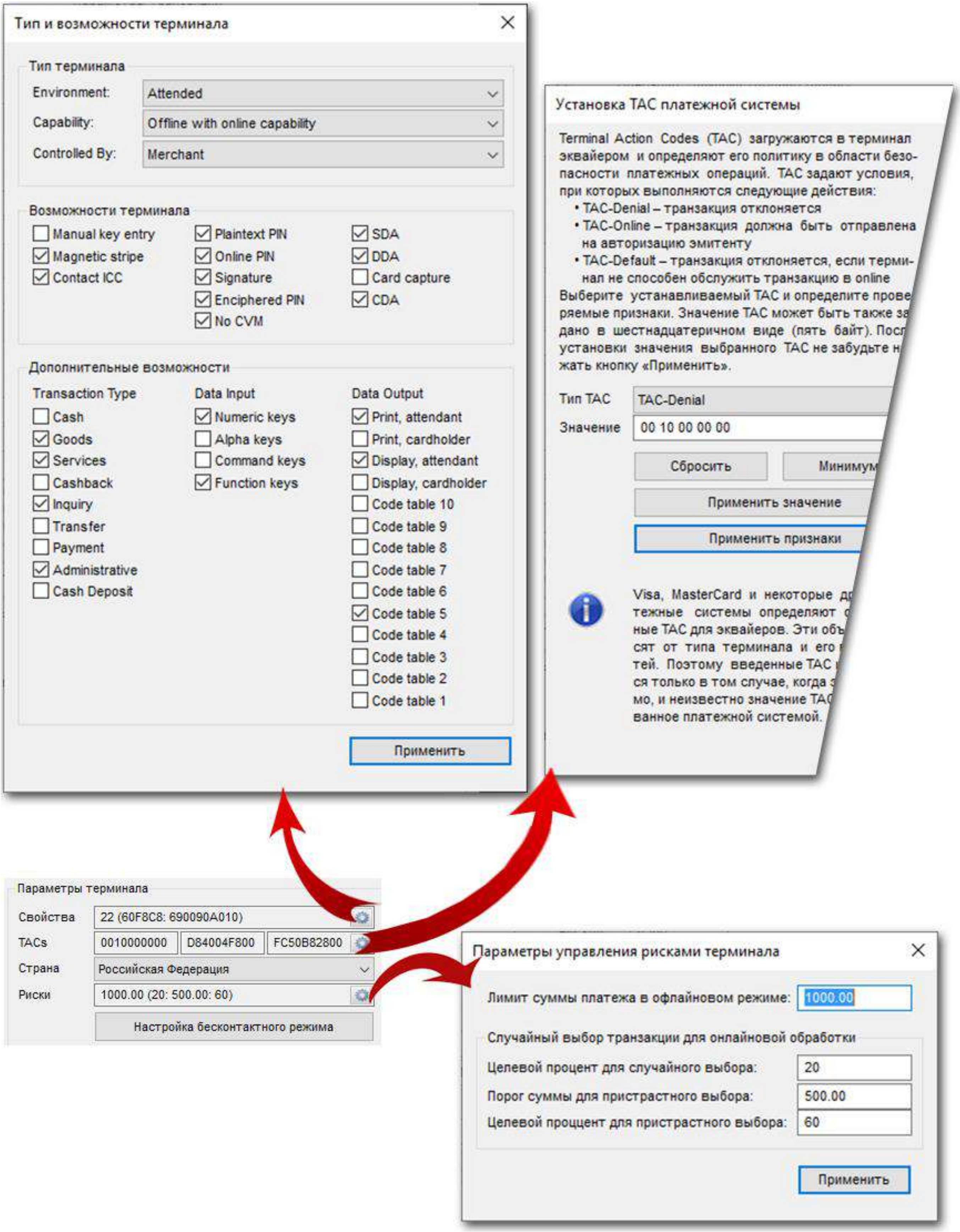


Рис. 9. Определение параметров терминала.

Определение ключей

На рис. 10 показаны средства комплекса тестирования, которые применяются для определения ключей. Эмулятор терминала использует публичные ключи платежных систем для выполнения транзакции. Ключи платежного приложения могут не использоваться (их применение опционально).

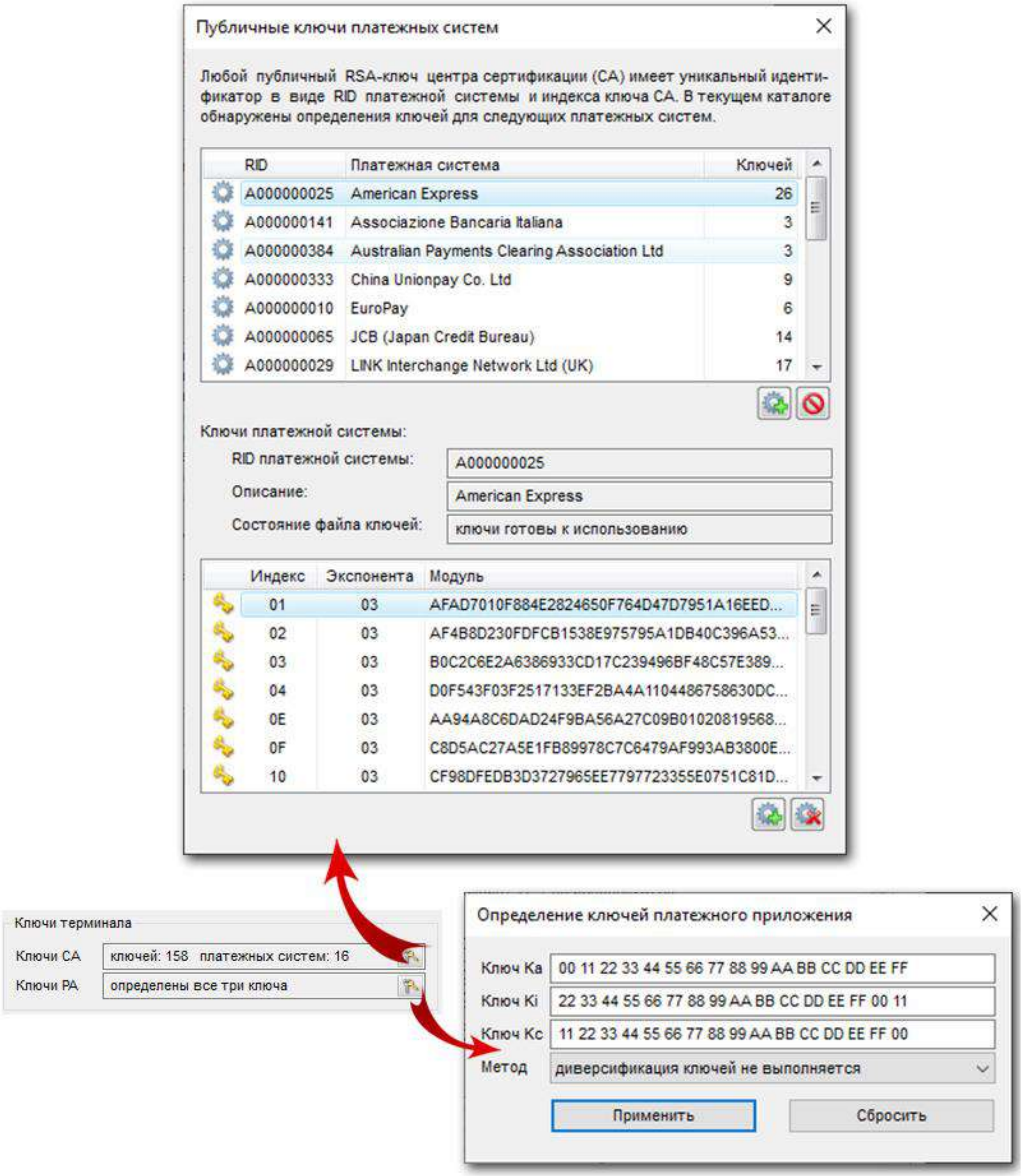


Рис. 10. Определение ключей.

Разберем, зачем терминалу нужны публичные ключи платежных систем (более точно – ключи центров сертификации платежных систем) для выполнения транзакции. Как уже описывалось ранее (см. раздел «Вопросы безопасности»), чтобы получить доступ к публичному RSA-ключу карты на первом этапе терминал должен восстановить публичный ключ эмитента из сертификата этого ключа, подписанного на секретном ключе центра сертификации (Certification Authority или CA). А зачем терминалу публичный RSA-ключ карты? Во-первых, для выполнения офлайновой аутентификации данных. Во-вторых, для зашифрования PIN-кода, если применяется метод верификации владельца карты «Предъявление зашифрованного PIN-кода карте». И отсутствие ключа центра сертификации платежной системы может повлиять (и очень значительно) на ход выполнения транзакции.

В связи с этим, в комплексе тестирования ECV существует база данных со всеми известными публичными ключами платежных систем. Но если даже какой-то платежной системы нет в базе, или для неё отсутствует нужный ключ, то это легко исправить. Предоставляется возможность создать новую платежную систему с определенным RID, или заблокировать все ключи определённой платежной системы (заблокированные ключи не удаляются из базы данных, но становятся недоступными эмулятору терминала). Кроме того, для любой платежной системы может быть создан новый ключ или удален существующий.

С ключами платежного приложения не так всё просто, как с публичными ключами платежных систем. Для выполнения транзакции они не нужны, но в определенных случаях могут потребоваться эмулятору терминала для выполнения следующих функций:

- проверки криптограммы транзакции (AAC, TC или ARQC), сгенерированной картой
- генерации криптограммы эмитента в ответ на запрос авторизации в случае эмуляции онлайновой обработки
- создания команд скрипт-процессинга, используемых эмитентом для модификации информации на карте (опять же, в случае эмуляции онлайновой обработки)

Хотя использование ключей платежного приложения в эмуляторе терминала заманчиво, но возможно не всегда – существует целый ряд ограничений. Во-первых, в эмуляторе реализованы только алгоритмы для приложения, созданного по спецификациям EMV CCD (Common Core Definitions). Во-вторых, сам процесс диверсификации ключей, выработки сессионных ключей и вычисления криптограмм, формирования команд скрипт-процессинга настолько многоэтапный и сложный, что для применения ключей платежного приложения в эмуляторе терминала нужна большая подготовка. Конечно, окончательное решение остается за пользователем комплекса тестирования.

Параметры транзакции

Группа управляющих элементов для ввода параметров платежной транзакции включает следующие объекты:

- combo-box для определения типа платежной операции (транзакции)
- сумму платежной операции
- сумму возврата наличными (используется только для покупки с возвратом наличными – cashback; во-всех остальных случаях должна быть равна 0)
- combo-box для определения валюты платежной операции

Кроме того, к этой группе управляющих элементов относится также строка редактирования для ввода PIN-кода владельца карты. Значение, указанное в качестве PIN-кода, используется только в том случае, когда в результате анализа списка методов верификации владельца карты установлено, что требуется предъявление PIN-кода карте. Если PIN-код не задан и требуется его предъявление карте, то считается, что владелец карты отказался от ввода PIN-кода.

Необходимо обратить внимание на следующие особенности применения PIN-кода эмулятором терминала.

1. Для онлайновой проверки значение PIN-кода не используется, так как эмулятор терминала не формирует данные для эмитента. Всегда считается, что предъявлен верный PIN-код.

2. Будьте осторожны и не забывайте, что PIN-код, подходящий для текущей карты, скорее всего будет неверен для следующей карты. Поэтому не забывайте менять значение PIN-кода для каждой анализируемой карты. Рекомендуется после проверки очередной карты удалять значение в строке редактирования для ввода PIN-кода.

Параметры эмуляции онлайновой обработки

На рис. 11 показано, каким образом определятся параметры для эмуляции онлайновой обработки в комплексе тестирования ECV.

Нужно сразу сказать, что онлайновая обработка моделируется только для контактного режима, так как в бесконтактном режиме эмулятор терминала всегда выполняет транзакцию за одно прикосновение к терминалу. После завершения работы с картой и получения от неё всех данных эмулятор считает, что обработка завершена. В реальном терминале анализируется ответ эмитента, принимается решение об одобрении или отклонении транзакции. Эти действия никогда не выполняются в эмуляторе терминала, потому что ничего нового для проверки платежного приложения они не дадут.

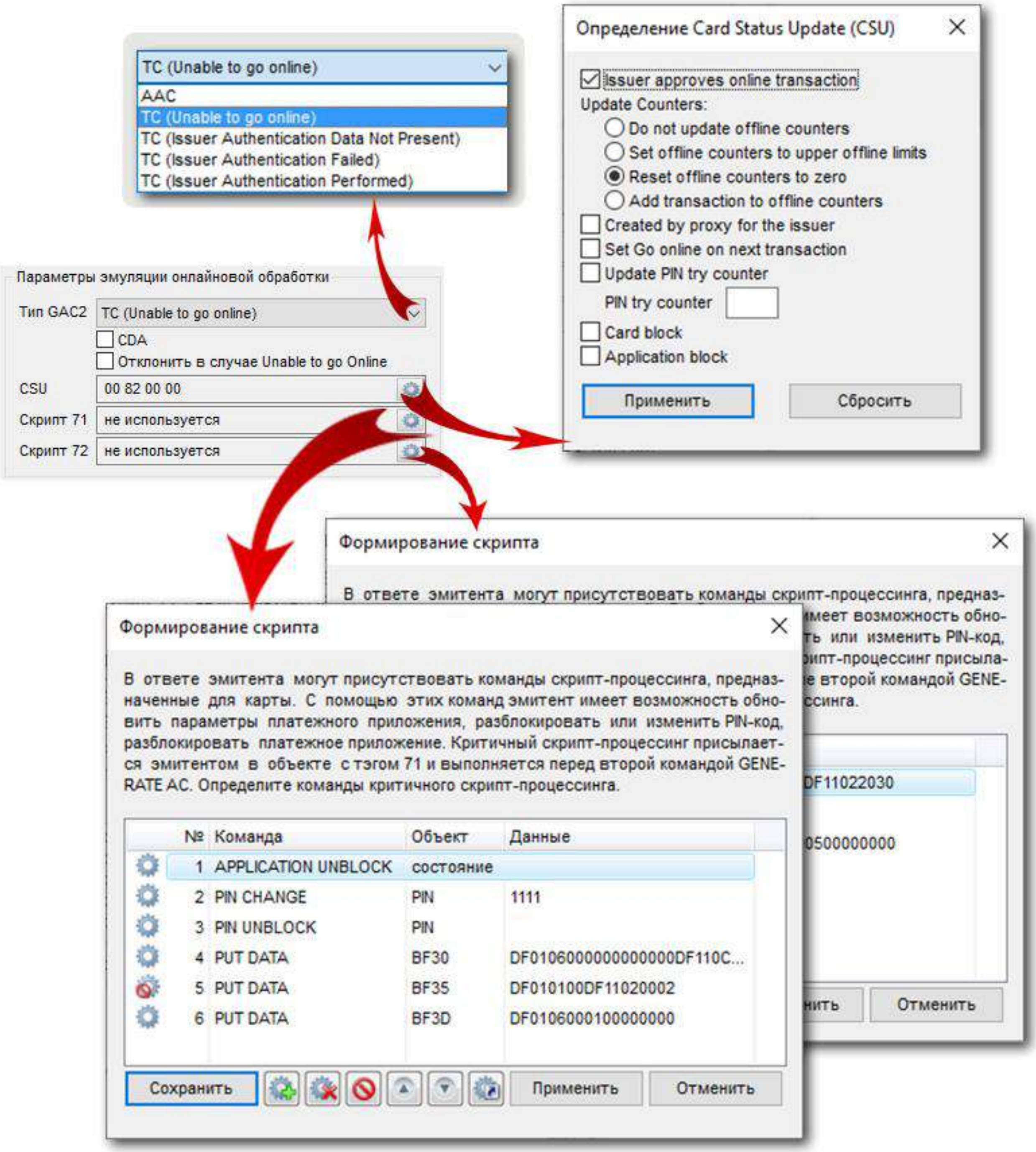


Рис. 11. Параметры эмуляции онлайновой обработки.

Для контактного режима эмуляция онлайнового режима имеет смысл, поскольку работа с картой после получения от неё всех данных для формирования авторизационного запроса эмитенту ещё не завершена. В реальной жизни терминал передает хосту обслуживающего банка сообщение, которое содержит информацию о транзакции. На основе этих данных хост обслуживающего банка формирует авторизационный запрос (сообщение x100 стандарта ISO 8583).

Но эмулятор терминала не предназначен для связи с обслуживающим банком и эмитентом. Никакие сообщения хосту обслуживающего банка или авторизационные запросы эмитенту не формируются. Вместо этого эмулятор может смоделировать различные варианты получения ответа эмитента (или его отсутствия).

Основным параметром для эмуляции онлайновой обработки является моделируемая ситуация, выбираемая из списка вариантов, определяемых combo-box «Тип GAC2»[1]. Из списка, определяемого этим элементом, выбирается один из следующих вариантов действий терминала при выполнении второй команды GENERATE AC:

- должна быть запрошена криптограмма AAC (отклонение транзакции)
- запрашивается криптограмма TC с моделированием ситуации «Unable to go Online»
- запрашивается криптограмма TC (имитируется ситуация «Данные эмитента не предоставлены»)
- запрашивается криптограмма TC (для платежного приложения моделируется ситуация «Аутентификация эмитента не выполнена»)
- во второй команде GENERATE AC запрашивается криптограмма TC и предоставляются данные эмитента, которые позволяют платежному приложению успешно выполнить аутентификацию эмитента (для приложения моделируется полноценный ответ эмитента)

Дальнейшее обсуждение особенностей перечисленных выше вариантов ещё предстоит, а пока нужно обсудить другие параметры эмуляции онлайнового режима. Сразу следует сказать, что эмуляция онлайнового режима была разработана для приложений, удовлетворяющих спецификациям EMV CCD (Common Core Definitions). И для целого ряда платежных приложений параметры онлайнового режима (объекты данных) не используются (не применимы).

Один из таких объектов данных – Card Status Update (CSU), который указывает, одобрена или отклонена транзакция эмитентом, а также определяет действия, которые с точки зрения эмитента должны быть выполнены картой (см. рис. 11). Определение признаков в CSU игнорируется, если хост эмитента не должен использовать CSU для управления платежным приложением.

Как показано на рис. 11, в комплексе тестирования ECV также предусмотрена возможность формирования команд критичного и некритичного скрипт-

[1] Имеется в виду тип запрашиваемой криптограммы во второй команде GENERATE AC. Из дальнейшего изложения будет ясно, что список определяет не только тип запрашиваемой криптограммы, но и различные варианты онлайновой обработки.

процессинга. Но команды скрипт-процессинга формируются в соответствии со спецификациями EMV CCD. Дело в том, что эмитент при создании команд скрипт-процессинга использует алгоритм Secure Messaging. Основные цели Secure Messaging – обеспечить целостность данных, предоставить возможность аутентификации источника данных и гарантировать конфиденциальность секретных данных. Алгоритм Secure Messaging для приложения EMV CCD реализован в соответствии с Secure Messaging Format 1, который описан в документе «EMV. Integrated Circuit Card Specifications for Payment Systems. Book 2. Security and Key Management. Version 4.2. June 2008».

Из этого следует два важных вывода. Во-первых, если для анализируемого платежного приложения команды скрипт-процессинга формируются не в соответствии с Secure Messaging Format 1, то средство определения команд скрипт-процессинга, предусмотренное в комплексе тестирования, использоваться не может. Во-вторых, как легко догадаться, для создания команд скрипт-процессинга эмулятору терминала требуются ключи платежного приложения (см. раздел «Определение ключей»).

А теперь возвращаемся к обсуждению вариантов действий эмулятора терминала при выполнении второй команды GENERATE AC. Как было описано выше, для эмуляции онлайновой обработки пользователь может выбрать один из пяти вариантов. Действия эмулятора терминала зависят от выбранного варианта. Но в любом случае, эмулятор выполняет следующее:

- перед выполнением второй команды GENERATE AC передает карте команды критичного скрипт-процессинга, если они определены

- после выполнения второй команды GENERATE AC посылает платежному приложению команды критичного скрипт-процессинга, если он используется

Остальные действия эмулятора терминала зависят от выбранного варианта развития событий.

Отклонение транзакции.

В этом случае действия эмулятора терминала совсем не интересны, так как платежное приложение игнорирует данные, предоставленные во второй команде GENERATE AC, и всегда возвращает криптограмму AAC (отклонение транзакции)

Одобрение транзакции в ситуации «Unable to go Online»

Эмулятор терминала сообщает платежному приложению об этой ситуации во второй команде GENERATE AC, запрашивая одобрение транзакции. Вместо данных эмитента обычно передаются нули.

Одобрение транзакции в ситуации «Данные эмитента не предоставлены»

Мало чем отличается от ситуации «Unable to go Online». Но есть отличия в данных, передаваемых платежному приложению во второй команде GENERATE AC, в связи с чем поведение платежного приложения может отличаться.

Одобрение транзакции и моделирование ситуации «Аутентификация эмитента не выполнена»

Эмулятор терминала запрашивает одобрение транзакции во второй команде GENERATE AC, передавая в качестве криптограммы эмитента случайное число (считается, что оно никогда не совпадет с криптограммой, которую должен сформировать эмитент в ответ на запрос авторизации).

Одобрение транзакции с полной эмуляцией ответа эмитента

Во второй команде GENERATE AC требуется одобрение транзакции с предоставлением данных, которые позволяют платежному приложению успешно выполнить аутентификацию эмитента и выполнить действия, запрашиваемые эмитентом. Для этого варианта эмулятору терминала требуются ключи платежного приложения (см. раздел «Определение ключей»).

Осталось только описать два переключателя, которые показаны на рис. 11. Переключатель «CDA» позволяет запросить от платежного приложения сертификат Signed Dynamic Application Data (используется для метода офлайновой аутентификации CDA), не только в первой, но и во второй команде GENERATE AC. Хотя на первый взгляд эта возможность и кажется излишней, но получение сертификата метода CDA во второй команде GENERATE AC определено спецификациями EMV для особых случаев. Таким образом, переключатель «CDA» позволяет проверить, способно ли платежное приложение предоставить сертификат Signed Dynamic Application Data во второй команде GENERATE AC.

Переключатель «Отклонить в случае Unable to go Online» отражает возможность, которая доступна обслуживающему банку для настройки терминала, работающего только в онлайновом режиме. Для таких терминалов может быть определено, что в ситуации Unable to go Online транзакция должна быть отклонена без анализа TAC/IAC.

Дополнительные проверки

Группа управляющих элементов, определяющих дополнительные проверки, которые выполняются в процессе анализа карты, позволяет осуществить следующие проверки:

- проверку PSE (Payment System Environment)
- анализ PPSE (Proximity Payment System Environment)
- отображение информации из журнала транзакций платежного приложения, если он поддерживается
- получение и анализ объектов по команде GET DATA

Далее приводится краткое описание этих дополнительных возможностей по проверке платежного приложения и его среды.

Проверка PSE

В контактном режиме для выбора платежного приложения может использоваться PSE (Payment System Environment) и комплекс тестирования позволяет выполнить его проверку. В терминологии EMV – это каталог, но на самом деле – это приложение с идентификатором 1PAY.SYS.DDF01. Эмулятор терминала сначала выбирает на карте приложение PSE, которое возвращает SFI файла со списком приложений на карте. Затем последовательно с помощью команды READ RECORD читает записи этого файла и извлекает из них информацию о платёжных приложениях, присутствующих на карте. Анализ PSE никогда не выполняется в бесконтактном режиме обработки.

Анализ PPSE

В бесконтактном режиме процедура выбора платежного приложения основана на использовании приложения PPSE (Proximity Payment System Environment), которое имеет идентификатор 2PAY.SYS.DDF0. В ответ на команду SELECT приложение PPSE возвращает объект FCI, содержащий информацию о бесконтактных приложениях на карте. Используя полученную информацию, эмулятор терминала проверяет список бесконтактных платёжных приложений, определённый в PPSE. Анализ PPSE никогда не выполняется в контактном режиме обработки.

Отображение информации из журнала транзакций

Ряд платежных приложений может иметь журнал транзакций – специальный файл, в котором регистрируются транзакции. Эмулятор терминала позволяет считать записи журнала транзакций и отобразить информацию из этих записей. Нужно иметь в виду, что журнал транзакций для любого платежного приложения является опциональным.

Получение объектов по команде GET DATA

Но не все данные платежного приложения расположены в записях файлов (записи файлов считываются по команде READ RECORD). Некоторые данные хранятся в виде отдельных объектов и при необходимости терминал извлекает их с карты с помощью команды GET DATA. Например, офлайновая проверка PIN-кода всегда начинается с того, что терминал выдает команду GET DATA для получения объекта PIN Try Counter. Значение этого объекта – это количество оставшихся попыток ввода PIN-кода.

Большинство объектов данных платежного приложения, считываемых с помощью команды GET DATA, для выполнения транзакции не нужны. Эмулятор терминала может считывать определенные объекты, чтобы информировать пользователя о том, почему карта приняла решение об обработке транзакции, отличающееся от решения терминала. Но эмулятор терминала никогда не считывает дополнительные, ненужные ему объекты, если не включён переключатель «Получение объектов по команде GET DATA». Установка этого переключателя заставляет эмулятор терминала считывать все известные ему объекты анализируемого платежного приложения. Информация из считанных объектов проверяется и объясняется.

Если включён переключатель «Получение объектов по команде GET DATA», то может быть установлен и переключатель «Поиск неизвестных объектов», информирующий о том, что требуется поиск всех объектов платежного приложения (даже не определенных в спецификациях), которые могут быть получены с помощью команды GET DATA. При установленном переключателе выполняется последовательное чтение всех объектов по тэгам, которые не определены в спецификациях анализируемого платежного приложения. Если тип платежного приложения не определен, то проверяется доступность всех объектов, кроме стандартных объектов EMV. Таких объектов много (больше 1000), поэтому не рекомендуется без особой нужды использовать данную возможность (поиск объектов может занять несколько десятков секунд). При выполнении транзакции в бесконтактном режиме в большинстве случаев установка этого переключателя игнорируется.

Журнал событий

Журнал событий, зафиксированных в процессе работы комплекса тестирования, – это основной источник информации о среде функционирования, результатах проверки карт, анализа данных и т. д. Журнал событий (протокол) отображается в отдельном окне программы и может быть просмотрен в любой момент работы с комплексом тестирования. Кроме того, протокол всегда сохраняется в текущем каталоге (каталоге, из которого запущена программа) в файле с именем Card Verification Log.x.y, где x – дата, а y – время создания файла с протоколом.

Обычно, журнал содержит много данных. Чтобы облегчить навигацию и поиск нужной информации, существуют кнопки управления журналом, показанные на рис. 12.



Рис. 12. Кнопки управления журналом.

Две кнопки со стрелочками позволяют перейти в окне протокола к началу исследования карты с платежным приложением. Например, предположим, что в окне отображается фрагмент исследования карты 3. Тогда нажатие кнопки со стрелкой вверх приведет к установке протокола в начало исследования карты 3, повторное нажатие – к установке в начало исследования карты 2 и т. д.

Аналогично, нажатие кнопки со стрелкой вниз приведет к установке протокола в начало исследования карты 4, повторное нажатие – к установке в начало исследования карты 5…

Последняя кнопка используется для сохранения протокола в файле и одновременной очистки окна протокола. Следует сразу сказать, что протокол в любом случае сохраняется в файле и обычно это делается в автоматическом режиме. Нажатие кнопки сохранения протокола приводит к следующему.

- Все строки, представленные в окне протокола, записываются в файл с именем Card Verification Log.x.y. Этот файл создается при запуске комплекса тестирования, и он всегда один (новый файл будет создан только при следующем запуске комплекса тестирования).

- Из окна протокола удаляются все строки (окно очищается).

Таким образом, кнопка сохранения протокола позволяет вывести из рассмотрения информацию, если сейчас она больше не нужна. Разумеется, эта информация не теряется и может быть проанализирована позже.

Кнопки управления

Для управления комплексом тестирования ECV предназначены кнопки, показанные на рис. 13.

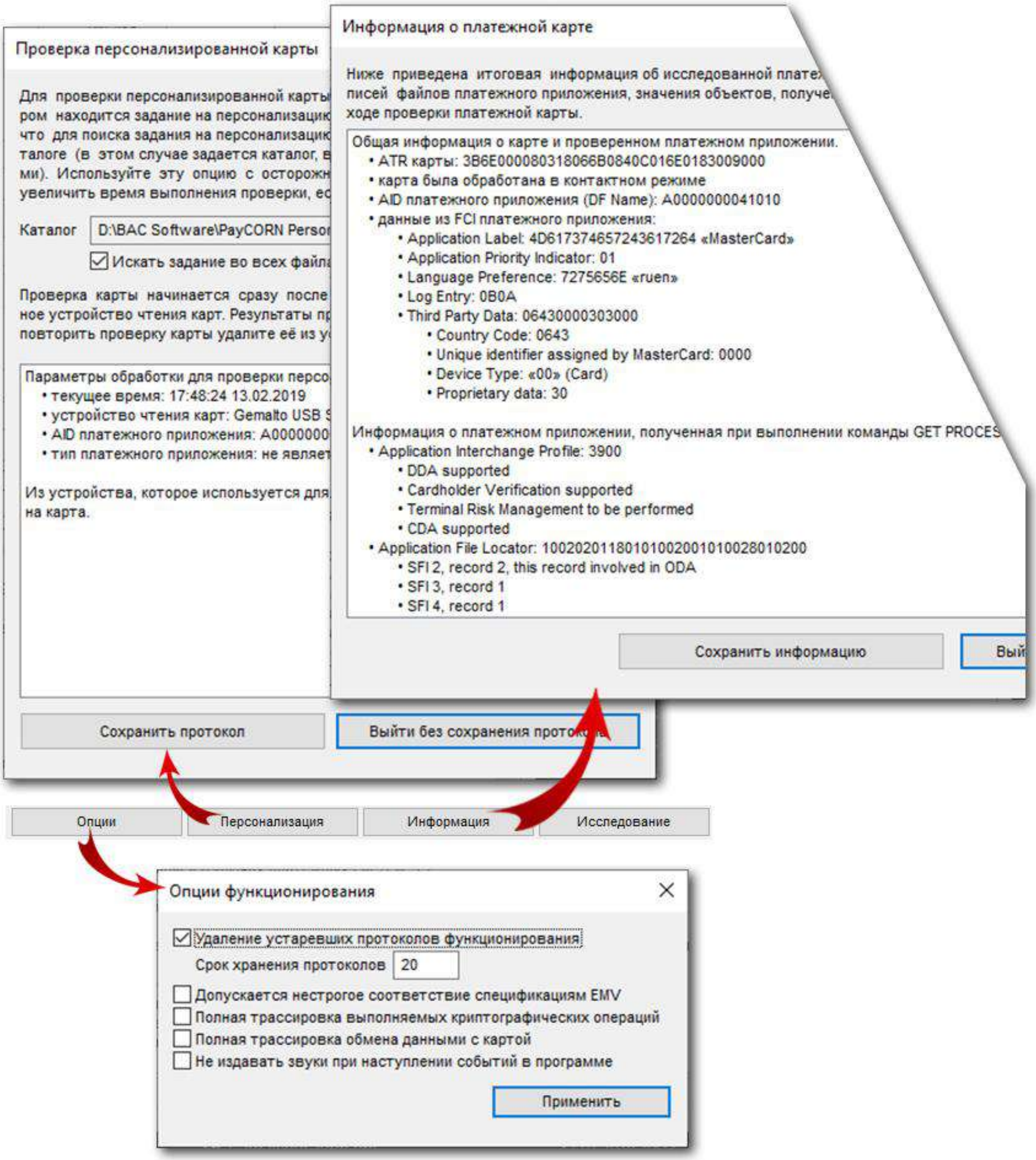


Рис. 13. Кнопки управления комплексом тестирования.

При нажатии на эти кнопки комплекс тестирования информируется, что пользователь желает выполнить определенное действие, связанное с определением параметров функционирования, проверкой карты или просмотром параметров платежного приложения. Далее подробно объясняется, какие действия выполняются при нажатии кнопок управления.

Опции

Кнопка «Опции» позволяет настроить опции функционирования комплекса тестирования, показанные на рис. 13 в окне «Опции функционирования». Среди настраиваемых опций есть сервисные возможности (например, удаление устаревших протоколов и запрет звуков при наступлении определенных событий), которые позволяют адаптировать комплекс тестирования под требования пользователя и никак не влияют на выполнение проверки платежных приложений.

Также присутствует довольно странная опция «Допускается неполное соответствие спецификациям EMV». На первый взгляд «вредная» возможность, но она позволяет продолжить проверку приложения, даже если уже обнаружены ошибки (например, это бывает полезно при проверке сертификатов ключей). Эту возможность не рекомендуется использовать неопытным пользователям, так как бывает достаточно трудно оценить, насколько то или иное несоответствие влияет на дальнейший ход тестирования.

Две оставшиеся возможности очень часто требуются при проверке платежного приложения. Во-первых, можно определить, что требуется полная трассировка выполняемых криптографических операций. Во-вторых, может быть запротоколирован весь обмен комплекса тестирования с картой.

Если эти возможности часто используются, то почему они являются опциональными? Ответ на этот вопрос достаточно прост. И трассировка выполняемых криптографических операций, и трассировка обмена данными с картой приводит к тому, что протокол проверки платежного приложения становится очень большим. Именно поэтому протокол, приведенный в документе (см. главу «Приложения»), создан без использования этих опций. Во многих случаях, дополнительные данные об обмене с картой и криптографических операциях не нужны. Как только пользователь принимает решение, что ему потребуются эти дополнительные данные, он всегда может включить опции дополнительных трассировок.

Когда определено, что выполнение криптографических операций должно быть занесено в протокол, то в журнале будет формироваться объяснение того, как комплекс тестирования реализует криптографические проверки. Например, на рис. 14 приведен фрагмент протокола с выполнением криптографической операции проверки сертификата публичного ключа эмитента. Для любой криптографической операции всегда приводятся исходные данные, промежуточные результаты и результаты выполнения (проверки).

```
Криптографическая операция проверки сертификата публичного ключа эмитента.
Исходные данные для операции:
  • экспонента публичного ключа CA: 03
  • модуль публичного ключа CA:
      B8 04 8A BC 30 C9 0D 97 63 36 54 3E 3F D7 09 1C
      8F E4 80 0D F8 20 ED 55 E7 E9 48 13 ED 00 55 5B
      57 3F EC A3 D8 4A F6 13 1A 65 1D 66 CF F4 28 4F
      B1 3B 63 5E DD 0E E4 01 76 D8 BF 04 B7 FD 1C 7B
      AC F9 AC 73 27 DF AA 8A A7 2D 10 DB 3B 8E 70 B2
      DD D8 11 CB 41 96 52 5E A3 86 AC C3 3C 0D 9D 45
      75 91 64 69 C4 E4 F5 3E 8E 1C 91 2C C6 18 CB 22
      DD E7 C3 56 8E 90 02 2E 6B BA 77 02 02 E4 52 2A
      2D D6 23 D1 80 E2 15 BD 1D 15 07 FE 3D C9 0C A3
      10 D2 7B 3E FC CD 8F 83 DE 30 52 CA D1 E4 89 38
      C6 8D 09 5A AC 91 B5 F3 7E 28 BB 49 EC 7E D5 97
  • зашифрованный сертификат ключа эмитента:
      20 8C 41 FC 26 54 76 24 9A 1D 72 B8 79 E1 61 4E
      9A C3 BD FC 1A EE C3 68 64 E2 6A 81 6E 85 E7 BD
      FA 3C 8F 34 BB 6F AC AD 0F 6B 90 D7 30 C2 5F A2
      EC B5 35 E2 3C CF 65 AD EC 07 9F 95 83 B7 65 BA
      C1 F8 E2 56 BD B9 5D A7 7A E9 56 1C 77 09 41 04
      B5 03 8B 2B C7 11 1A 43 76 87 43 F5 50 3F 60 8D
      DD 17 9C F0 46 CB B5 9F 2C 4F 65 7A 87 13 D3 96
      C5 58 5B 95 60 A9 2B 3D 57 FD 23 96 01 A1 D4 84
      FB E9 26 EF 3C 41 3F 57 AE BE 2E 8D F5 02 D9 00
      2B 5E 2F 1B 79 B8 99 22 74 1E 26 32 EE A9 E3 07
      FE 7F 27 35 3F 8F 7C 77 EE 3B ED A9 2A A9 B7 9D
Результаты криптографической операции проверки сертификата публичного ключа эмитента:
  • расшифрованный сертификат ключа эмитента:
      6A 02 53 45 26 FF 12 23 00 9A 6F 01 01 90 01 F2
      82 6F 7C E3 B5 0F 5C FF 50 EC 40 25 72 DF 70 EE
      87 E5 09 7A 1C 40 BF 0A 2D 57 25 BD A8 E4 EC 05
      60 B7 81 DE F1 78 71 93 2E 86 89 50 AB 4F A5 A4
      6C 5A EA 27 5C 60 60 D4 7E 23 08 E8 BD F8 46 54
      E9 DC 2B F3 C3 40 27 00 3B F7 DA 2A 7A 7E 3B 97
      54 BC 6F 91 0A A3 9C 5B 69 D4 A7 E0 C1 94 07 27
      9F 6C 6C FC BA 6E 10 6D B6 76 4E 49 E9 25 90 F7
      8B DB F8 A9 8A E0 4D 6C 8F 15 11 76 0F 59 BF 89
      A7 AB D9 9A 60 38 44 EC 7D BC F4 9A 37 A5 C7 50
      7E 0B 62 89 77 59 A5 F0 2F 9C EF CA 54 67 31 BC
  • проверяется хэш сертификата, для чего используются следующие дополнительные данные:
      • Issuer Public Key Remainder: 421490D7
      • Issuer Public Key Exponent: 03
```

Рис. 14. Трассировка криптографических операций.

Если задана трассировка обмена данными с картой, то в протокол будут помещаться информация о командах, передаваемые карте, и ответе, полученном от карты. На рис. 15 приведен фрагмент протокола с включенной трассировкой обмена данными с картой (строки, поясняющие работу с картой, выделены красным цветом). Для любой команды отображается её кодировка, данные, передаваемые вместе с командой, а также данные, полученные от карты и байты состояния (код возврата карты). Следует иметь в виду, что трассировка действий с картой не зависит от любой другой трассировки. Поэтому в протоколе могут появляться избыточные данные (например, на рис. 15 ответ карты на команду GET PROCESSING OPTIONS приведен в двух местах).

7. Выполнение команды GET PROCESSING OPTIONS для инициирования транзакции и получения информации, необходимой для выполнения транзакции.
- для инициирования транзакции никакие данные не нужны, поскольку не определен PDOL, и в качестве входных данных команды предоставляется объект Command Template (тег 83) с нулевой длиной
- платежному приложению передается команда GET PROCESSING OPTIONS

Действие с картой:
- команда: 80 A8 00 00
- длина данных для карты: 2
- данные для карты: 83 00
- код карты: 61 10

Действие с картой:
- команда: GET RESPONSE
- ожидаемая длина данных от карты: 16
- длина данных, полученных от карты: 16
- данные от карты: 77 0E 82 02 38 00 94 08 28 01 03 01 30 01 02 00
- код карты: 90 00

- команда GET PROCESSING OPTIONS завершена успешно
- время выполнения команды: 203 мсек
- в ответ на команду получены следующие данные: 770E82023800940828010301300102 00
- интерпретация полученной TLV-структуры::
    - 77.14 Response Message Template Format 2
        - 82.2 Application Interchange Profile
        - 94.8 Application File Locator
- выполняется анализ данных, полученных в ответ на команду GET PROCESSING OPTIONS
- команда предоставила следующие данные:
    - Application Interchange Profile: 3800
        - DDA supported
        - Cardholder Verification supported
        - Terminal Risk Management to be performed
    - Application File Locator: 2801030130010200
        - SFI 5, records 1 - 3, record 1 involved in ODA
        - SFI 6, records 1 - 2

Рис. 15. Трассировка обмена данными с картой.

Персонализация

Кнопка «Персонализация» предназначена для запуска процесса, с помощью которого можно проверить правильность персонализации платежного приложения по заданию на выпуск карты (файлу с данными для персонализации EMV-приложения), сформированного подсистемой подготовки данных персонализации из входных данных бэк-офиса банка-эмитента.

Применение этой функции ограничено прежде всего тем, что для проверки правильности персонализации платежного приложения требуется XML-файл, подготовленный подсистемой персонализации OpenWay (Way4 Smart Card Perso) для внешнего персо-бюро. С подсистемами персонализации других провайдеров персо-решений комплекс тестирования не работает.

Кроме того, в полной мере проверка персонализации осуществляется только для платежного приложения PayCORN. Для других типов платежных приложений может быть выполнена только частичная проверка.

Информация

При нажатии кнопки «Информация» на экране отображается окно с обобщенной информацией о последней проверенной карте. На рис. 16 показан фрагмент такой информации. Предоставляются следующие данные:

- общая информация о карте и платежном приложении
- результат выполнения команды GET PROCESSING OPTIONS
- данные, считанные из файлов платежного приложения
- данные, считанные с помощью команды GET DATA
- некоторые дополнительные данные о платежном приложении.

Общая информация о карте и проверенном платежном приложении.
- ATR карты: 3B6E000080318066B0840C016E0183009000
- карта была обработана в контактном режиме
- AID платежного приложения (DF Name): A0000000041010
- данные из FCI платежного приложения:
    - Application Label: 4D617374657243617264 «MasterCard»
    - Application Priority Indicator: 01
    - Language Preference: 7275656E «ruen»
    - Log Entry: 0B0A
    - Third Party Data: 06430000303000
        - Country Code: 0643
        - Unique identifier assigned by MasterCard: 0000
        - Device Type: «00» (Card)
        - Proprietary data: 30

Информация о платежном приложении, полученная при выполнении команды GET PROCESSING OPTIONS.
- Application Interchange Profile: 3900
    - DDA supported
    - Cardholder Verification supported
    - Terminal Risk Management to be performed
    - CDA supported
- Application File Locator: 10020201180101002001010028010200
    - SFI 2, record 2, this record involved in ODA
    - SFI 3, record 1
    - SFI 4, record 1
    - SFI 5, records 1 - 2

Данные, считанные из файлов платежного приложения.
- Application PAN: 5225980034347618
- Application PAN Sequence Number: 00
- Application Effective Date: 01.10.2018
- Application Expiration Date: 30.11.2021
- Application Version Number: 0002
- Application Currency Code: 0643
- Cardholder Name: 5349444F524F562F564C4144494D4952202020202020202020202020 'SIDOROV/VLADIMIR '
- Issuer Country Code (numeric): 0643
- Application Usage Control: FF00
    - valid for domestic cash transactions
    - valid for international cash transactions
    - valid for domestic goods
    - valid for international goods
    - valid for domestic services
    - valid for international services
    - valid at ATMs
    - valid at terminals other than ATMs

Рис. 16. Информация о проверенной карте.

Обобщенная информация о последней проверенной карте может быть просмотрена и сохранена в текущем каталоге программы в файле с именем Card Info.x.y, где x – дата, а y – время создания файла.

Нужно обратить внимание на то, что обобщенная информация – это всего лишь выборка из тех данных, которые представлены в протоколе, итоги проверки платежного приложения.

Если ещё не было обработано ни одно платежное приложение, то никакая информация не отображается.

Исследование

Кнопка «Исследование» инициирует процесс проверки заданного платежного приложения, находящегося на установленной карте. Прежде чем начать анализ платежного приложения, необходимо сделать следующее.

1. Определить, каким образом будет осуществляться поиск платежного приложения на карте. Возможно, установить тип приложения.

2. Установить среду проверки платежного приложения (задать параметры терминала, транзакции и эмуляции онлайновой обработки).

3. Выбрать устройство чтения смарт-карт, в которое будет установлена карта с проверяемым приложением.

4. Установить карту с проверяемым приложением в выбранное устройство.

После нажатия кнопки «Исследование» проверка платежного приложения выполняется в автоматическом режиме и никакие действия от пользователя больше не требуются.

Глава

3

Приложения

Эта глава содержит сведения справочного характера, которые могут помочь в понимании возможностей комплекса тестирования ECV и избежать определенных ошибок при проверке платежных приложений. Ссылки на эти сведения присутствуют в других главах документа, но только в этой главе они представлены и объяснены в полном объеме.

В данной главе приведено описание некоторых криптографических алгоритмов, которые применяются картой, терминалом и эмитентом. Все эти алгоритмы подробно описаны в спецификациях EMV. Здесь алгоритмы приведены, так как упоминались в тексте документа и являются общими для всех платежных приложений. Описание алгоритмов может использоваться только для справки и не должно применяться в других целях. При описании криптографических методов используются следующие соглашения.

1. Для определения операции зашифрования данных по алгоритму Triple DES используется обозначение: 3DESECB (Data, K) – зашифрование данных Data на ключе K в режиме ECB.

2. Для определения операций зашифрования и расшифрования данных с использованием алгоритма RSA используются следующие обозначения:

   - RSARPUB (Sign, Kp) – восстановление данных из подписи Sign на открытом ключе Kp (т. е. расшифрование данных на открытом ключе)

   - RSASPRV (Data, Ks) – получение подписи данных Data на секретном ключе Ks (т. е. зашифрование данных на секретном ключе)

Диверсификация мастер-ключа

При вводе ключей платежного приложения (см. раздел «Определение ключей») существует три метода преобразования введенных ключи в ключи платежного приложения:

- диверсификация (преобразование) ключей не выполняется
- диверсификация ключей по опции A
- диверсификация ключей по опции B

В последних двух случаях считается, что в качестве ключей определены мастер-ключи эмитента, которые должны быть диверсифицированы в ключи платежного приложения.

В данном разделе описан алгоритм диверсификации мастер-ключа эмитента (IMK) с использованием PAN (Primary Account Number) и PAN Sequence Number. В результате такой диверсификации вычисляется ключ CMK, уникальный для платежного приложения. В соответствии с документом «EMV. Integrated Circuit Card Specifications for Payment Systems. Book 2. Security and Key Management. Version 4.2. June 2008» эмитент может использовать две опции для диверсификации мастер-ключа – опцию A и опцию B.

Процесс диверсификации включает следующие шаги.

1. Если PAN содержит больше 16-ти десятичных цифр и используется опция B диверсификации ключа, то выполняется переход к шагу 2. Иначе, за десятичными цифрами PAN располагаются цифры PAN Sequence Number (если PAN Sequence Number не задан, то вместо него используется нулевой байт). Если результат X содержит меньше 16-ти цифр, то он дополняется нулями слева, чтобы получить 8-ми байтное число Y. Когда результат X содержит по крайней мере 16 цифр, 8-ми байтное число Y представляет собой 16 самых правых цифр результата X. Выполняется переход к шагу 3.

2. Если PAN содержит нечетное количество цифр, то он дополняется нулем слева. Затем за десятичными цифрами PAN располагаются цифры PAN Sequence Number (если PAN Sequence Number не задан, то вместо него используется нулевой байт). Вычисляется хэш-функция полученного значения по алгоритму SHA-1, в результате чего будет получен 20-ти байтный результат X. Затем выбираются первые 16 десятичных цифр из результата X, чтобы получить 8-ми байтное число Y. Если в числе Y недостаточно десятичных цифр, шестнадцатеричные цифры X преобразуются в десятичные путем вычитания 10. Например, если X = '`1230ABCD567842D4B179F2CA345D6789A17B64BB`', то значение Y = '`1230567842417923`' (первые 16 десятичных цифр результата X).

Если X = '1B3CABCDD6E8FAD4B1CDF2CAD4FDC78FA17B6EBB', то значение Y = '1368412478176'. К этим десятичным цифрам затем добавляется результат преобразования шестнадцатеричных цифр '120' (полученных из 'B', 'C' и 'A') и число Y = '1368412478176120'.

3. Вычисляется ключ CMK как результат выполнения следующих операций:

$$Z_1 = 3DESECB\ (Y, IMK)$$

$$Z_2 = 3DESECB\ (Y \oplus 0xFFFFFFFFFFFFFFFF, IMK)$$

$$CMK = Z_1 \mid\mid Z_2$$

Таким образом, опция A диверсификации мастер-ключа является подмножеством опции B. Эмитент может использовать любую из опций, поскольку они абсолютно равнозначны. Разумеется, опция A более проста в реализации, а опция B – более продвинутая.

Восстановление публичного ключа эмитента

Для целого ряда действий с платежным приложением (выполнения офлайновой аутентификации данных, предъявления зашифрованного PIN-кода) терминал должен иметь публичный ключ карты. Чтобы получить публичный ключ карты из данных платежного приложения, терминал сначала должен восстановить публичный ключ эмитента из сертификата публичного ключа эмитента, подписанного на секретном ключе Certification Authority (CA). Далее приведен алгоритм этого процесса.

Терминал выполняет следующие шаги для проверки сертификата публичного ключа эмитента.

1. Проверяет длину сертификата публичного ключа эмитента (она должна равняться длине модуля публичного ключа CA - Nca).

2. Расшифровывает сертификат публичного ключа эмитента на публичном ключе CA (Pca) по формуле RSARPUB (Certificate, Pca). Расшифрованный сертификат должен иметь следующий вид.

Смещение	Длина	Содержимое
0	1	Заголовок сертификата (0x6A).
1	1	Идентификатор формата (0x02)
2	4	Идентификатор эмитента (от 3-х до 8-ми самых левых цифр PAN, дополненных при необходимости шестнадцатеричными цифрами F справа)
6	2	Дата истечения срока действия сертификата (n4 в виде MMYY)
8	3	Серийный номер сертификата
11	1	Идентификатор хэш-алгоритма (0x01 – SHA1)
12	1	Идентификатор алгоритма генерации сертификата (0x01 – RSA)
13	1	Длина модуля публичного ключа эмитента (Ni)
14	1	Длина экспоненты публичного ключа эмитента (1 - 3)
15	Nca – 36	Старшие (самые левые) байты модуля публичного ключа эмитента[1]
Nca - 21	20	Значение хэш-функции для публичного ключа эмитента и связанной с ним информации
Nca - 1	1	Идентификатор окончания сертификата (0xBC)

[1] Если Ni <= Nca - 36, то сертификат содержит весь модуль публичного ключа эмитента, дополненный справа байтами 0xBB (количество байт дополнения равно Nca - Ni - 36). Иначе, сертификат содержит Nca - 36 старших байт модуля публичного ключа эмитента (остаток публичного ключа эмитента заносится в объект Issuer Public Key Remainder).

3. Проверяет идентификатор окончания сертификата (должен быть равен 0xBC), заголовок сертификата (должен быть равен 0x6A) и идентификатор формата (должен быть равен 0x02).

4. Получает значение хэш-функции по алгоритму SHA1 для конкатенации следующих элементов данных:

   - идентификатор формата (0x02)
   - идентификатор эмитента
   - дата истечения срока действия сертификата
   - серийный номер сертификата
   - идентификатор хэш-алгоритма (0x01)
   - идентификатор алгоритма генерации сертификата (0x01)
   - длина модуля публичного ключа эмитента (Ni)
   - длина экспоненты публичного ключа эмитента (1 – 3)
   - модуль публичного ключа эмитента (старшие байты модуля публичного ключа эмитента из сертификата, за которыми следуют младшие байты модуля из объекта данных Issuer Public Key Remainder, полученного от карты, или модуль публичного ключа эмитента из сертификата, дополненный справа байтами 0xBB, если весь модуль поместился в сертификате)
   - экспонента публичного ключа эмитента

5. Сравнивает полученное значение хэш-функции со значением, определённым в сертификате.

6. Проверяет, что идентификатор эмитента соответствует первым цифрам PAN (при этом учитывается, что цифры идентификатор эмитента может содержать от 3-х до 8-ми самых левых цифр PAN, дополненных при необходимости шестнадцатеричными цифрами F справа).

7. Проверяет, что срок действия сертификата не истек.

8. Проверяет идентификатор алгоритма генерации сертификата (должен быть равен 0x01)

Если любая из перечисленных проверок не выполнена, то считается, что аутентификация карты провалилась. Иначе, сертификат публичного ключа верен и модуль публичного ключа эмитента извлекается из сертификата или получается путем конкатенации старших байтов модуля публичного ключа эмитента из сертификата и младших байтов модуля из объекта данных Issuer Public Key Remainder.

Восстановление публичного ключа карты

Для выполнения офлайновой аутентификации данных и предъявления зашифрованного PIN-кода терминал должен восстановить публичный ключ карты из сертификата публичного ключа карты, подписанного на секретном ключе эмитента. Для этого требуется публичный ключ эмитента. Процедура восстановления публичного ключа эмитента подробно описана в предыдущем разделе. После восстановления публичного ключа эмитента терминал восстанавливает публичный ключ карты, используя его сертификат. Терминал выполняет следующие шаги для проверки сертификата публичного ключа карты.

1. Проверяет длину сертификата публичного ключа карты (она должна равняться длине модуля публичного ключа эмитента – Ni).

2. Расшифровывает сертификат публичного ключа карты на публичном ключе эмитента (Pi) по формуле RSARPUB (Certificate, Pi). Расшифрованный сертификат должен иметь следующий вид.

Смещение	Длина	Содержимое
0	1	Заголовок сертификата (0x6A).
1	1	Идентификатор формата (0x04)
2	10	Application PAN (дополненный справа шестнадцатеричными цифрами F)
12	2	Дата истечения срока действия сертификата (n4 в виде MMYY)
14	3	Серийный номер сертификата
17	1	Идентификатор хэш-алгоритма (0x01 – SHA1)
18	1	Идентификатор алгоритма генерации сертификата (0x01 – RSA)
19	1	Длина модуля публичного ключа карты (Nic)
20	1	Длина экспоненты публичного ключа карты (1 – 3)
21	Ni - 42	Старшие (самые левые) байты модуля публичного ключа карты[1]
Ni - 21	20	Значение хэш-функции для публичного ключа карты и связанной с ним информации
Ni - 1	1	Идентификатор окончания сертификата (0xBC)

[1] Если Nic <= Ni - 42, то сертификат содержит весь модуль публичного ключа карты, дополненный справа байтами 0xBB (количество байт дополнения равно Ni - Nic - 42). Иначе, сертификат содержит Ni - 42 старших байт модуля публичного ключа карты (остаток публичного ключа карты заносится в объект ICC Public Key Remainder)

3. Проверяет идентификатор окончания сертификата (должен быть равен 0xBC), заголовок сертификата (должен быть равен 0x6A) и идентификатор формата (должен быть равен 0x04).

4. Получает значение хэш-функции по алгоритму SHA1 для конкатенации следующих элементов данных:

   - идентификатор формата (0x02)
   - Application PAN
   - дата истечения срока действия сертификата
   - серийный номер сертификата
   - идентификатор хэш-алгоритма (0x01)
   - идентификатор алгоритма генерации сертификата (0x01)
   - длина модуля публичного ключа карты (Nic)
   - длина экспоненты публичного ключа карты (1 – 3)
   - модуль публичного ключа карты (старшие байты модуля публичного ключа карты из сертификата, за которыми следуют младшие байты модуля из объекта данных ICC Public Key Remainder, полученного от карты, или модуль публичного ключа карты из сертификата, дополненный справа байтами 0xBB, если весь модуль поместился в сертификате)
   - экспонента публичного ключа карты
   - статические данные, которые должны быть аутентифицированы (могут отсутствовать)

   Статические данные, которые должны быть аутентифицированы, определяются элементами списка AFL (Application File Locator) в том порядке, в котором они появляются в списке AFL и считываются терминалом. Данные, включаемые в процесс аутентификации, зависят от Short File Identifier (SFI) файла, из которого считываются записи.

   - для файлов с SFI в диапазоне от 1 до 10 тэг записи (70) и длина записи исключаются из процесса аутентификации. Все другие элементы данных из записи включаются.
   - для файлов с SFI в диапазоне от 11 до 30 тэг записи (70) и длина записи, а также другие элементы данных включаются в процесс аутентификации.

После того как все элементы, определяемые AFL, включены в состав статической информации, которая должна быть аутентифицирована, обрабатывается Static Data Authentication Tag List, если он определен в данных, считанных терминалом по команде READ RECORD. Static Data Authentication Tag List, если он задан, может содержать только тэг для Application Interchange Profile (AIP). Таким образом, если элемент данных Static Data Authentication Tag List определен, то значение AIP заносится в конец статических данных, которые должны быть аутентифицированы (тэг и длина AIP не включаются).

5. Сравнивает полученное значение хэш-функции со значением, определённым в сертификате.

6. Проверяет, что Application PAN, определенный в сертификате, соответствует Application PAN, полученном от платежного приложения.

7. Проверяет, что срок действия сертификата не истек.

8. Проверяет идентификатор алгоритма генерации сертификата (должен быть равен 0x01)

Если любая из перечисленных проверок не выполнена, то считается, что офлайновая аутентификация данных провалилась. Иначе, сертификат публичного ключа верен и модуль публичного ключа карты извлекается из сертификата или получается путем конкатенации старших байтов модуля публичного ключа карты из сертификата и младших байтов модуля из объекта данных ICC Public Key Remainder.

Метод CDA

Метод офлайновой аутентификации данных, который называется CDA (Combined Data Authentication) сейчас наиболее распространен для карточных продуктов. Это самый сложный из методов офлайновой аутентификации, в связи с чем анализ платежного приложения, использующего метод CDA, может вызвать затруднения. В связи с этим приводится описание операций, которые должны выполнить карта и терминал, чтобы обеспечить офлайновую аутентификацию данных по методу CDA.

Подпись CDA (сертификат, предоставляемый в объекте Signed Dynamic Application Data), генерируется картой по определенному алгоритму. В процессе генерации сертификата выполняются следующие действия.

Сначала формируются данные сертификата Signed Dynamic Application Data. Данные представляются в виде поля фиксированной длины, размер которого равен длине ключа карты (Nic). Поле имеет следующий формат.

Смещение	Длина	Содержимое
0	1	Заголовок сертификата (0x6A).
1	1	Идентификатор формата (0x05)
2	1	Идентификатор хэш-алгоритма (0x01 – SHA1)
3	1	Длина динамических данных в байтах (38). В таблице динамические данные выделены цветом.
4	1	Длина ICC Dynamic Number (8)
5	8	ICC Dynamic Number
13	1	Cryptogram Information Data (CID)
14	8	Криптограмма (TC или ARQC)
22	20	Transaction Data Hash Code
42	Nic - 63	Байты со значением 0xBB
Nic - 21	20	Dynamic Application Data Hash
Nic - 1	1	Идентификатор окончания сертификата (0xBC)

Необходимо дать несколько пояснений к элементам данным, используемым платежным приложением для генерации сертификата.

1. ICC Dynamic Number – это криптографическая функция от значения ATC, которая определяется разработчиком платежного приложения.

2. Transaction Data Hash Code – значение хэш-функции по алгоритму SHA1 для конкатенации следующих элементов данных:

    - значения элементов, указанных в PDOL[1]
    - значения элементов, указанных в CDOL1
    - значения элементов, указанных в CDOL2 (только для второй команды GENERATE AC – для первой команды GENERATE AC это поле опущено)
    - объекта Cryptogram Information Data с тэгом 9F27 и длиной 1, объекта ATC с тэгом 9F36 и длиной 2, объекта Issuer Application Data с тэгом 9F10 и длиной 32[2]

3. Dynamic Application Data Hash – значение хэш-функции по алгоритму SHA1 для конкатенации следующих элементов данных:

    - идентификатор формата (0x05)
    - идентификатор хэш-алгоритма (0x01)
    - длина динамических данных в байтах (38)
    - динамические данные
    - байты со значением 0xBB (длиной Nic - 63)
    - 4-х байтное случайное число терминала, переданное карте в списке CDOL1 или CDOL2

Подготовленные данные подписываются (зашифровываются) на секретном ключе карты (ICC Private Key – Sicc) по формуле RSASPRV (Data, Sicc) в результате чего формируется сертификат Signed Dynamic Application Data.

Терминал, получив от карты сертификат Signed Dynamic Application Data, выполняет следующие шаги для проверки предоставленного сертификата (аутентификации карты).

1. Проверяет длину предоставленного сертификата (она должна равняться длине модуля ICC Public Key – Nic).

2. Расшифровывает сертификат на публичном ключе карты (ICC Public Key – Picc) по формуле RSARPUB (Certificate, Picc).

[1] Если список PDOL не используется, то поле должно быть опущено.

[2] Это объекты, которые возвращены в ответ на команду GENERATE AC в том порядке, в котором они представлены в ответе (за исключением объекта Signed Dynamic Application Data).

3. Проверяет идентификатор окончания сертификата (должен быть равен 0xBC), заголовок сертификата (должен быть равен 0x6A) и идентификатор формата (должен быть равен 0x05).

4. Сравнивает Cryptogram Information Data из сертификата со значением Cryptogram Information Data, возвращенным командой GENERATE AC.

5. Вычисляет Dynamic Application Data Hash по тому же алгоритму, что и карта, и сравнивает полученное значение со значением, определённым в сертификате.

6. Вычисляет Transaction Data Hash Code по тому же алгоритму, что и карта, и сравнивает полученное значение со значением, определённым в сертификате.

Если любая из перечисленных проверок не выполнена, то считается, что аутентификация карты провалилась.

Пример протокола

В этом разделе приведен пример протокола исследования платежного приложения, удовлетворяющего спецификациям MasterCard. Рекомендуется обратить внимание на следующие особенности исследования:

- транзакция выполнена в контактном режиме
- успешно выполнена офлайновая аутентификация данных карты с использованием метода CDA
- выполнен метод верификации владельца карты «Предъявление PIN-кода для передачи его эмитенту», так как предъявление PIN-кода карте провалилось из-за неправильного значения PIN-кода
- выданы команды GET DATA для получения информации об объектах платежного приложения
- первая команда GENERATE AC вернула криптограмму ARQC и была выполнена эмуляция онлайновой обработки
- выдана вторая команда GENERATE AC для завершения транзакции.

Инициировано исследование установленной карты с платёжным приложением. В процессе анализа карты будут использоваться следующие параметры:

- тип терминала: 22
  - Attended, Offline with online capability
  - Operational control provided by Merchant
- возможности терминала:
  - Card data input capability: Magnetic stripe, IC with contacts
  - CVM capability: Plaintext PIN for ICC verification, Enciphered PIN for online verification, Signature (paper), Enciphered PIN for offline verification, No CVM allowed
  - Security capability: SDA, DDA, CDA
- расширенные возможности терминала:
  - Transaction type capability: Goods, Services, Inquiry, Administrative
  - Transaction data input capability: Numeric keys, Function keys
  - Transaction data output capability: Print (attendant), Display (attendant)
  - Code tables: 5 (Latin/Cyrillic: кириллица, включающая символы славянских языков)
- страна, в которой расположен терминал: Российская Федерация
- параметры процедуры управления рисками терминала (Terminal Risk Management):
  - максимальное значение суммы платежа в офлайновом режиме (Terminal Floor Limit): 1000.00
  - целевой процент, используемый в процедуре случайного выбора транзакции для онлайновой обработки: 20
  - пороговое значение суммы платежа для пристрастного выбора, используемое в процедуре случайного выбора транзакции для онлайновой обработки: 500.00
  - максимальный целевой процент пристрастного выбора, используемый в процедуре случайного выбора транзакции для онлайновой обработки: 60
- тип транзакции: 00 (покупка товаров или услуг)
- сумма платежной операции: 20.00
- другая сумма (сумма cashback): 0.00
- валюта платежной операции: русский рубль
- дата транзакции: 12.02.2019
- время транзакции: 20:07:37

При проверке платежной карты выполняются следующие обязательные шаги и опциональные действия, запланированные пользователем.

1. Первоначальный анализ установленной карты.
    - ATR карты: 3B 6E 00 00 80 31 80 66 B0 84 0C 01 6E 01 83 00 90 00
    - предполагается контактный режим
    - протокол: T0
2. Установка проверяемого платежного приложения в качестве текущего приложения на карте (операция, с которой начинается любая платежная транзакция).
    - осуществляется холодный сброс карты, чтобы исключить побочные эффекты предыдущих действий
    - установка текущего приложения с помощью команды SELECT
    - в ответ на команду получены следующие данные:
      6F 33 84 07 A0 00 00 00 04 10 10 A5 28 50 0A 4D
      61 73 74 65 72 43 61 72 64 5F 2D 04 72 75 65 6E
      87 01 01 BF 0C 0F 9F 4D 02 0B 0A 9F 6E 07 06 43
      00 00 30 30 00
    - интерпретация полученной TLV-структуры:
        - 6F.51 FCI Template
            - 84.7 Dedicated File Name
            - A5.40 FCI Proprietary Template
                - 50.10 Application Label
                - 5F2D.4 Language Preference
                - 87.1 Application Priority Indicator
                - BF0C.15 FCI Issuer Discretionary Data
                    - 9F4D.2 Log Entry
                    - 9F6E.7 Third Party Data
    - выполняется анализ данных, полученных в ответ на команду SELECT (анализ FCI платежного приложения)
    - в FCI платежного приложения найдены следующие объекты, которые могут использоваться при обработке транзакции:
        - Dedicated File Name: A0000000041010
        - Application Label: 4D617374657243617264 'MasterCard'
        - Application Priority Indicator: 01
        - Language Preference: 7275656E 'ruen'
        - Log Entry: 0B0A

        - Third Party Data: 06430000303000
            - Country Code: 0643
            - Unique identifier assigned by MasterCard: 0000
            - Device Type: «00» (Card)
            - Proprietary data: 30
    - платежное приложение будет обрабатываться в соответствии со спецификациями MasterCard

3. Получение значений объектов платежного приложения, определенных в общих спецификациях EMV и детальных спецификациях платежного приложения (с использованием команды GET DATA).
    - выдача команды GET DATA для получения значения объекта Application Transaction Counter (ATC)
        - значение объекта платежного приложения не получено (объект отсутствует в платежном приложении)
    - выдача команды GET DATA для получения значения объекта Last Online ATC Register
        - значение объекта платежного приложения не получено (объект отсутствует в платежном приложении)
    - выдача команды GET DATA для получения значения объекта PIN Try Counter
        - время выполнения команды: 16 мсек
        - значение объекта сохраняется для дальнейшей обработки
    - выдача команды GET DATA для получения значения объекта Log Format
        - время выполнения команды: 46 мсек
        - значение объекта сохраняется для дальнейшей обработки
    - выдача команды GET DATA для получения значения объекта Card Issuer Action Code – Decline
        - время выполнения команды: 31 мсек
        - значение объекта сохраняется для дальнейшей обработки
    - выдача команды GET DATA для получения значения объекта Card Issuer Action Code – Default
        - время выполнения команды: 31 мсек
        - значение объекта сохраняется для дальнейшей обработки
    - выдача команды GET DATA для получения значения объекта Card Issuer Action Code – Online
        - время выполнения команды: 31 мсек
        - значение объекта сохраняется для дальнейшей обработки
    - выдача команды GET DATA для получения значения объекта Counters
        - время выполнения команды: 31 мсек
        - значение объекта сохраняется для дальнейшей обработки
    - выдача команды GET DATA для получения значения объекта CDOL1 Related Data Length
        - время выполнения команды: 32 мсек
        - значение объекта сохраняется для дальнейшей обработки

- выдача команды GET DATA для получения значения объекта Card Risk Management Country Code
  - время выполнения команды: 32 мсек
  - значение объекта сохраняется для дальнейшей обработки
- выдача команды GET DATA для получения значения объекта Card Risk Management Currency Code
  - время выполнения команды: 31 мсек
  - значение объекта сохраняется для дальнейшей обработки
- выдача команды GET DATA для получения значения объекта Lower Cumulative Offline Transaction Amount
  - время выполнения команды: 31 мсек
  - значение объекта сохраняется для дальнейшей обработки
- выдача команды GET DATA для получения значения объекта Upper Cumulative Offline Transaction Amount
  - время выполнения команды: 31 мсек
  - значение объекта сохраняется для дальнейшей обработки
- выдача команды GET DATA для получения значения объекта Card Issuer Action Code (Contactless) – Default
  - время выполнения команды: 31 мсек
  - значение объекта сохраняется для дальнейшей обработки
- выдача команды GET DATA для получения значения объекта Card Issuer Action Code (Contactless) – Online
  - время выполнения команды: 31 мсек
  - значение объекта сохраняется для дальнейшей обработки
- выдача команды GET DATA для получения значения объекта Card Issuer Action Code (Contactless) – Decline
  - время выполнения команды: 31 мсек
  - значение объекта сохраняется для дальнейшей обработки
- выдача команды GET DATA для получения значения объекта Currency Conversion Table
  - время выполнения команды: 62 мсек
  - значение объекта сохраняется для дальнейшей обработки
- выдача команды GET DATA для получения значения объекта Additional Check Table
  - время выполнения команды: 47 мсек
  - значение объекта сохраняется для дальнейшей обработки
- выдача команды GET DATA для получения значения объекта Application Control
  - время выполнения команды: 31 мсек
  - значение объекта сохраняется для дальнейшей обработки
- выдача команды GET DATA для получения значения объекта Default ARPC Response Code
  - время выполнения команды: 32 мсек
  - значение объекта сохраняется для дальнейшей обработки

- выдача команды GET DATA для получения значения объекта Application Control (Contactless)
  - время выполнения команды: 31 мсек
  - значение объекта сохраняется для дальнейшей обработки
- выдача команды GET DATA для получения значения объекта Lower Consecutive Offline Limit
  - время выполнения команды: 31 мсек
  - значение объекта сохраняется для дальнейшей обработки
- выдача команды GET DATA для получения значения объекта Upper Consecutive Offline Limit
  - время выполнения команды: 31 мсек
  - значение объекта сохраняется для дальнейшей обработки
- выдача команды GET DATA для получения значения объекта Offline Balance
  - значение объекта платежного приложения не получено (объект недоступен в данном контексте)
- выдача команды GET DATA для получения значения объекта Data Recovery DOL
  - значение объекта платежного приложения не получено (объект отсутствует в платежном приложении)
- выдача команды GET DATA для получения значения объекта Application Life Cycle Data
  - время выполнения команды: 79 мсек
  - значение объекта сохраняется для дальнейшей обработки
- выдача команды GET DATA для получения значения объекта Security Limits Status
  - время выполнения команды: 31 мсек
  - значение объекта сохраняется для дальнейшей обработки

4. Проверка и интерпретация значений объектов платежного приложения, считанных с помощью команды GET DATA.
   - анализ данных, полученных от платежного приложения
     - PIN Try Counter: 3 (03)
     - Log Format: 9F27019F02065F2A029A039F36029F5206
       - 9F27.1 Cryptogram Information Data (CID)
       - 9F02.6 Amount, Authorized (numeric)
       - 5F2A.2 Transaction Currency Code
       - 9A.3 Transaction Date
       - 9F36.2 Application Transaction Counter (ATC)
       - 9F52.6 Card Verification Results (CVR)
     - Card Issuer Action Code – Decline: 000000
     - Card Issuer Action Code – Default: 195000
       - Offline PIN verification failed
       - Pin try limit exceeded

- Terminal erroneously considers offline PIN OK
- Upper consecutive offline limit exceeded
- Upper cumulative offline limit exceeded
- Card Issuer Action Code – Online: 39FB00
  - Offline PIN verification not performed
  - Offline PIN verification failed
  - Pin try limit exceeded
  - Terminal erroneously considers offline PIN OK
  - Lower consecutive offline limit exceeded
  - Upper consecutive offline limit exceeded
  - Lower cumulative offline limit exceeded
  - Upper cumulative offline limit exceeded
  - Go online on next transaction was set
  - Script received
  - Script failed
- Counters: 00380000000000000000
  - ATC: 0038
  - Global MAC in Script Counter: 0
  - Bad Cryptogram Counter: 0
- CDOL1 Related Data Length: 43 (2B)
- Card Risk Management Country Code: 0643
- Card Risk Management Currency Code: 0643
- Lower Cumulative Offline Transaction Amount: 1500.00
- Upper Cumulative Offline Transaction Amount: 1600.00
- Card Issuer Action Code (Contactless) – Default: 005800
  - Upper consecutive offline limit exceeded
  - Upper cumulative offline limit exceeded
  - Go online on next transaction was set
- Card Issuer Action Code (Contactless) – Online: 00F800
  - Lower consecutive offline limit exceeded
  - Upper consecutive offline limit exceeded
  - Lower cumulative offline limit exceeded
  - Upper cumulative offline limit exceeded

    - Go online on next transaction was set
- Card Issuer Action Code (Contactless) – Decline: 080000
    - Pin try limit exceeded
- Currency Conversion Table: 0643000000064300000006430000000643000000064300000000
    - Currency code: 0643
    - Conversion factor: not defined
    - Currency code: 0643
    - Conversion factor: not defined
    - Currency code: 0643
    - Conversion factor: not defined
    - Currency code: 0643
    - Conversion factor: not defined
    - Currency code: 0643
    - Conversion factor: not defined
- Additional Check Table: 000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
    - There is a format error in the Additional Check Table:
- Application Control: 8C00
    - Magnetic Stripe grade issuer activated (allows the card to accept transactions when Issuer Authentication data is not present)
    - Offline encrypted PIN verification supported
    - ICC key for offline encrypted PIN verification
    - Offline plaintext PIN verification supported
    - MasterCard proprietary session key derivation
- Default ARPC Response Code: 0010
    - PIN Try Counter: 0
    - Approve online transaction
    - Do not update PIN Try Counter
    - Reset go online on next transaction
    - Update counters: do not update offline counters
- Application Control (Contactless): 000080
    - Magnetic Stripe grade issuer not activated
    - MasterCard proprietary session key derivation
    - Use static CVC3 (PayPass)

- Lower Consecutive Offline Limit: 05
- Upper Consecutive Offline Limit: 06
- Application Life Cycle Data:
  03 10 19 12 00 09 00 00 A1 A2 A3 A4 A5 A6 A7 A8
  A9 AA AB AC AD AE AF B0 B1 B2 B3 B4 C1 C2 C3 C4
  C5 C6 C7 C8 C9 CA CB CC CD CE CF D0 D1 D2 D3 D4
  - Version: M/Chip Select 4
  - Type Approval ID: 10191200090000
  - Application Issuer ID: A1A2A3A4...B1B2B3B4
  - Application Code ID: C1C2C3C4...D1D2D3D4
- Security Limits Status: 00

5. Выполнение команды GET PROCESSING OPTIONS для инициирования транзакции и получения информации, необходимой для выполнения транзакции.
   - для инициирования транзакции никакие данные не нужны, поскольку не определен PDOL, и в качестве входных данных команды предоставляется объект Command Template (тег 83) с нулевой длиной
   - платежному приложению передается команда GET PROCESSING OPTIONS
   - команда GET PROCESSING OPTIONS завершена успешно
   - время выполнения команды: 78 мсек
   - в ответ на команду получены следующие данные: 771682023900941010020201180101002001010028010200
   - интерпретация полученной TLV-структуры:
     - 77.22 Response Message Template Format 2
       - 82.2 Application Interchange Profile
       - 94.16 Application File Locator
   - выполняется анализ данных, полученных в ответ на команду GET PROCESSING OPTIONS
   - команда предоставила следующие данные:
     - Application Interchange Profile: 3900
       - DDA supported
       - Cardholder Verification supported
       - Terminal Risk Management to be performed
       - CDA supported
     - Application File Locator: 10020201180101002001010028010200
       - SFI 2, record 2, this record involved in ODA
       - SFI 3, record 1

      - SFI 4, record 1
      - SFI 5, records 1 - 2

6. Чтение данных из записей файлов платежного приложения.
   - выдается команда READ RECORD для чтения записи 2 из файла с идентификатором 2
   - время выполнения команды: 172 мсек
   - в ответ на команду получены следующие данные:

     70 81 8C 5A 08 52 25 98 00 34 34 76 18 5F 24 03
     21 11 30 5F 25 03 18 10 01 5F 28 02 06 43 5F 34
     01 00 8C 21 9F 02 06 9F 03 06 9F 1A 02 95 05 5F
     2A 02 9A 03 9C 01 9F 37 04 9F 35 01 9F 45 02 9F
     4C 08 9F 34 03 8D 0C 91 0A 8A 02 95 05 9F 37 04
     9F 4C 08 8E 14 00 00 00 00 00 00 00 00 42 01 44
     03 41 03 42 03 1E 03 1F 03 9F 07 02 FF 00 9F 0D
     05 BC 50 BC 88 00 9F 0E 05 00 00 00 00 00 9F 0F
     05 BC 70 BC 98 00 9F 42 02 06 43 9F 4A 01 82

   - интерпретация полученной TLV-структуры:
     - 70.140 READ RECORD Template
       - 5A.8 Application PAN
       - 5F24.3 Application Expiration Date
       - 5F25.3 Application Effective Date
       - 5F28.2 Issuer Country Code (numeric)
       - 5F34.1 Application PAN Sequence Number
       - 8C.33 Card Risk Management DOL 1 (CDOL1)
       - 8D.12 Card Risk Management DOL 2 (CDOL2)
       - 8E.20 CVM List
       - 9F07.2 Application Usage Control
       - 9F0D.5 IAC-Default
       - 9F0E.5 IAC-Denial
       - 9F0F.5 IAC-Online
       - 9F42.2 Application Currency Code
       - 9F4A.1 Static Data Authentication Tag List
   - объекты из считанной записи сохраняются для дальнейшей обработки (все элементарные объекты данных из записи будут использоваться для офлайновой аутентификации данных)

- выдается команда READ RECORD для чтения записи 1 из файла с идентификатором 3
- время выполнения команды: 266 мсек
- в ответ на команду получены следующие данные:
  70 81 E0 8F 01 05 90 81 B0 94 ED 79 BD 06 7B 12
  46 39 D0 89 1E B3 CF EA AC 5A A1 44 9F 45 09 ED
  3E C5 E1 BD 99 AC EF 5B 01 4D C8 02 60 55 C4 55
  6A 97 01 62 D8 AC 61 29 A5 F8 1F 0E 11 86 2E 02
  05 E1 AD 18 BB 98 12 39 88 2D 22 35 58 8D 68 4A
  59 25 18 01 BA 74 DB C0 C9 59 4A ED 35 D2 E6 41
  9F E1 C2 80 BE 69 63 61 16 B8 6F BC B8 64 4A E4
  5B 83 69 37 49 9B 6C 74 52 9E FE FC DC D8 D9 8A
  76 55 CE 63 C3 E3 91 E5 50 F9 B5 F1 31 F1 C5 7A
  48 E7 B4 ED D4 C5 30 4B 99 1F 16 6C CA E5 7C 6F
  EA 91 CA 65 6E 20 20 EA C7 14 6D F4 EB DA 48 1B
  42 46 30 F3 92 3F 61 70 47 92 24 C9 BC 26 10 83
  1D F1 A6 A7 DB A2 E9 E6 33 40 1A 54 0F 40 57 BD
  56 49 F8 E3 15 8E 2C 03 0A 22 3C 45 B6 F7 ED 9F
  32 01 03
- интерпретация полученной TLV-структуры:
  - 70.224 READ RECORD Template
    - 8F.1 CA Public Key Index
    - 90.176 Issuer Public Key Certificate
    - 92.36 Issuer Public Key Remainder
    - 9F32.1 Issuer Public Key Exponent
- объекты из считанной записи сохраняются для дальнейшей обработки
- выдается команда READ RECORD для чтения записи 1 из файла с идентификатором 4
- время выполнения команды: 31 мсек
- в ответ на команду получены следующие данные: 70049F470103
- интерпретация полученной TLV-структуры:
  - 70.4 READ RECORD Template
    - 9F47.1 ICC Public Key Exponent
- объекты из считанной записи сохраняются для дальнейшей обработки

- выдается команда READ RECORD для чтения записи 1 из файла с идентификатором 5
- время выполнения команды: 93 мсек
- в ответ на команду получены следующие данные:
  70 47 9F 1F 0D 31 30 31 38 36 30 30 30 30 30 32
  30 38 57 13 52 25 98 00 34 34 76 18 D2 11 12 01
  10 18 60 00 00 20 8F 5F 20 1A 53 49 44 4F 52 4F
  56 2F 56 4C 41 44 49 4D 49 52 20 20 20 20 20 20
  20 20 20 20 9F 08 02 00 02
- интерпретация полученной TLV-структуры:
  - 70.71 READ RECORD Template
    - 9F1F.13 Track 1 Discretionary Data
    - 57.19 Track 2 Equivalent Data
    - 5F20.26 Cardholder Name
    - 9F08.2 Application Version Number
- объекты из считанной записи сохраняются для дальнейшей обработки
- выдается команда READ RECORD для чтения записи 2 из файла с идентификатором 5
- время выполнения команды: 219 мсек
- в ответ на команду получены следующие данные:
  70 81 BA 9F 46 81 B0 91 7D 25 34 98 48 68 E3 4D
  43 52 09 74 8C 5F 91 2C B1 80 18 E2 74 83 53 20
  87 6F B6 75 DD 2D C2 71 93 5A 71 08 E5 A3 AD 1D
  63 D7 D1 69 BF FF 83 20 25 39 D7 9D F8 99 E2 B7
  69 05 F4 68 39 16 C6 1D 6E 3A AB F6 56 D4 CF 65
  5D 7A C6 B2 D9 4A 55 30 59 44 66 BD B8 EA 53 80
  06 80 6F A9 F3 81 91 B1 06 9B 73 10 E8 E5 95 62
  19 8C 39 60 59 50 73 72 A4 E0 06 52 07 BF B1 66
  5B FC 64 60 EE CB D5 AE 3D B8 99 B0 70 7A F6 AA
  70 D8 E6 9E A9 07 CD 1C D8 FB 3E B0 8F E2 64 31
  0A 1D 58 91 97 DC 60 24 C0 3E 1A 59 D4 10 E8 3D
  7F 69 08 DC 01 6B 03 9F 49 03 9F 37 04
- интерпретация полученной TLV-структуры:
  - 70.186 READ RECORD Template
    - 9F46.176 ICC Public Key Certificate

        - 9F49.3 Dynamic Data Authentication DOL (DDOL)
    - объекты из считанной записи сохраняются для дальнейшей обработки

7. Анализ и интерпретация данных, считанных из файлов платежного приложения, в соответствии с общими спецификациями EMV.
    - известные объекты, сохраненные в базе данных терминала
        - Application PAN: 5225980034347618
        - Application PAN Sequence Number: 00
        - Application Effective Date: 01.10.2018
        - Application Expiration Date: 30.11.2021
        - Application Version Number: 0002
        - Application Currency Code: 0643
        - Cardholder Name: 5349444F524F562F564C4144494D495220202020202020202020 'SIDOROV/VLADIMIR '
        - Issuer Country Code (numeric): 0643
        - Application Usage Control: FF00
            - valid for domestic cash transactions
            - valid for international cash transactions
            - valid for domestic goods
            - valid for international goods
            - valid for domestic services
            - valid for international services
            - valid at ATMs
            - valid at terminals other than ATMs
        - CVM List: 000000000000000042014403410342031E031F03
            - Amount X = 0
            - Amount Y = 0
            - 4201 (enciphered PIN verified online, if unattended cash, apply succeeding rule)
            - 4403 (enciphered PIN verification performed by ICC, if terminal supports CVM, apply succeeding rule)
            - 4103 (plaintext PIN verification performed by ICC, if terminal supports CVM, apply succeeding rule)
            - 4203 (enciphered PIN verified online, if terminal supports CVM, apply succeeding rule)
            - 1E03 (signature, if terminal supports CVM, fail cardholder verification)
            - 1F03 (no CVM required, if terminal supports CVM, fail cardholder verification)

- Card Risk Management DOL 1 (CDOL1):
  9F 02 06 9F 03 06 9F 1A 02 95 05 5F 2A 02 9A 03
  9C 01 9F 37 04 9F 35 01 9F 45 02 9F 4C 08 9F 34
  03
  - 9F02.6 Amount, Authorized (numeric)
  - 9F03.6 Amount, Other (numeric)
  - 9F1A.2 Terminal Country Code
  - 95.5 Terminal Verification Results
  - 5F2A.2 Transaction Currency Code
  - 9A.3 Transaction Date
  - 9C.1 Transaction Type
  - 9F37.4 Unpredictable Number
  - 9F35.1 Terminal Type
  - 9F45.2 Data Authentication Code (DAC)
  - 9F4C.8 ICC Dynamic Number
  - 9F34.3 CVM Results
- Card Risk Management DOL 2 (CDOL2): 910A8A0295059F37049F4C08
  - 91.10 Issuer Authentication Data
  - 8A.2 Authorization Response Code
  - 95.5 Terminal Verification Results
  - 9F37.4 Unpredictable Number
  - 9F4C.8 ICC Dynamic Number
- IAC-Default: BC50BC8800
  - offline data authentication was not performed
  - ICC data missing
  - card appears on terminal exception file
  - DDA failed
  - CDA failed
  - expired application
  - requested service not allowed for card product
  - cardholder verification was not successful
  - PIN try limit exceeded
  - PIN required and PIN pad not present or not working

- PIN required, PIN pad present, but PIN was not entered
- online PIN entered
- transaction exceeds floor limit
- merchant forced transaction online

- IAC-Denial: 0000000000
- IAC-Online: BC70BC9800
  - offline data authentication was not performed
  - ICC data missing
  - card appears on terminal exception file
  - DDA failed
  - CDA failed
  - expired application
  - application not yet effective
  - requested service not allowed for card product
  - cardholder verification was not successful
  - PIN try limit exceeded
  - PIN required and PIN pad not present or not working
  - PIN required, PIN pad present, but PIN was not entered
  - online PIN entered
  - transaction exceeds floor limit
  - transaction selected randomly for online processing
  - merchant forced transaction online
- CA Public Key Index: 5 (05)
- Issuer Public Key Certificate:

94 ED 79 BD 06 7B 12 46 39 D0 89 1E B3 CF EA AC
5A A1 44 9F 45 09 ED 3E C5 E1 BD 99 AC EF 5B 01
4D C8 02 60 55 C4 55 6A 97 01 62 D8 AC 61 29 A5
F8 1F 0E 11 86 2E 02 05 E1 AD 18 BB 98 12 39 88
2D 22 35 58 8D 68 4A 59 25 18 01 BA 74 DB C0 C9
59 4A ED 35 D2 E6 41 9F E1 C2 80 BE 69 63 61 16
B8 6F BC B8 64 4A E4 5B 83 69 37 49 9B 6C 74 52
9E FE FC DC D8 D9 8A 76 55 CE 63 C3 E3 91 E5 50
F9 B5 F1 31 F1 C5 7A 48 E7 B4 ED D4 C5 30 4B 99

1F 16 6C CA E5 7C 6F EA 91 CA 65 6E 20 20 EA C7
14 6D F4 EB DA 48 1B 42 46 30 F3 92 3F 61 70 47

- Issuer Public Key Remainder:
  C9 BC 26 10 83 1D F1 A6 A7 DB A2 E9 E6 33 40 1A
  54 0F 40 57 BD 56 49 F8 E3 15 8E 2C 03 0A 22 3C
  45 B6 F7 ED
- Issuer Public Key Exponent: 03
- ICC Public Key Certificate:
  91 7D 25 34 98 48 68 E3 4D 43 52 09 74 8C 5F 91
  2C B1 80 18 E2 74 83 53 20 87 6F B6 75 DD 2D C2
  71 93 5A 71 08 E5 A3 AD 1D 63 D7 D1 69 BF FF 83
  20 25 39 D7 9D F8 99 E2 B7 69 05 F4 68 39 16 C6
  1D 6E 3A AB F6 56 D4 CF 65 5D 7A C6 B2 D9 4A 55
  30 59 44 66 BD B8 EA 53 80 06 80 6F A9 F3 81 91
  B1 06 9B 73 10 E8 E5 95 62 19 8C 39 60 59 50 73
  72 A4 E0 06 52 07 BF B1 66 5B FC 64 60 EE CB D5
  AE 3D B8 99 B0 70 7A F6 AA 70 D8 E6 9E A9 07 CD
  1C D8 FB 3E B0 8F E2 64 31 0A 1D 58 91 97 DC 60
  24 C0 3E 1A 59 D4 10 E8 3D 7F 69 08 DC 01 6B 03
- ICC Public Key Exponent: 03
- Dynamic Data Authentication DOL (DDOL): 9F3704
  - 9F37.4 Unpredictable Number
- Static Data Authentication Tag List: 82
  - 82 Application Interchange Profile
- Track 2 Equivalent Data: 5225980034347618D21112011018600000208F
  - Primary Account Number: 5225980034347618
  - Field separator: D
  - Expiration Date: 11.2021
  - Service code: 201 (International, use chip where feasible; Normal transaction authorization; No restrictions)
  - Discretionary data: 1018600000208
  - Pad to ensure whole bytes: F
- Track 1 Discretionary Data: 31303138363030303030323038 '1018600000208'

8. Офлайновая аутентификация данных карты в соответствии с возможностями терминала и карты.
   - карта сообщает, что поддерживаются методы DDA и CDA офлайновой аутентификации
   - терминал поддерживает методы SDA, DDA И CDA офлайновой аутентификации
   - для офлайновой аутентификации данных будет использоваться метод CDA
   - выполняется восстановление публичного ключа эмитента
   - осуществляется поиск публичного ключа платежной системы, используемого для проверки сертификата ключа эмитента, по следующим параметрам:
     - RID платежной системы: A000000004
     - индекс ключа платежной системы (CA Public Key Index): 5 (05)
     - длина модуля ключа платежной системы: 176
   - найден единственный подходящий публичный ключ платежной системы, с помощью которого проверяется сертификат публичного ключа эмитента
   - сертификат публичного ключа эмитента признан достоверным, из него извлечены срок действия сертификата и публичный ключ эмитента:
     - срок действия сертификата: 12.2022
     - экспонента публичного ключа: 03
     - модуль публичного ключа:

       AD E0 57 07 5A D8 60 50 04 8A 4D 53 3C 01 05 4E
       D2 CB 84 83 56 93 D4 D1 23 04 25 3D E5 D7 D6 7C
       07 6F 01 FF 33 4F FC 46 35 39 8C 7B 70 EE 32 61
       34 B6 76 F6 5B D0 66 F3 AC 3C 4C 43 CE 68 1D 37
       BD DD 15 38 7C 6D A3 5E 4F 56 4F B9 66 F8 56 A8
       53 85 79 E0 4B 28 82 61 10 BC 49 AB 97 48 AB
       F8 C1 2B B8 26 51 86 78 52 6C 60 CE 16 FB 2C 6D
       73 C2 A6 1B B1 33 F0 49 FA 13 44 F9 99 3F 58 A9
       0C 37 86 ED E9 10 97 89 38 E1 84 96 C9 BC 26 10
       83 1D F1 A6 A7 DB A2 E9 E6 33 40 1A 54 0F 40 57
       BD 56 49 F8 E3 15 8E 2C 03 0A 22 3C 45 B6 F7 ED
   - проверяется сертификат публичного ключа карты
   - сертификат публичного ключа карты признан достоверным, из него извлечены срок действия сертификата и публичный ключ карты:
     - срок действия сертификата: 11.2021
     - экспонента публичного ключа: 03

  - модуль публичного ключа:
    B6 DF E9 32 85 59 70 45 3E 8D 39 35 19 F9 FA 51
    F3 8A 54 C2 64 D1 35 67 2E 32 31 76 2C CF EF 43
    3C A4 C5 96 E1 6E BD CB 55 20 EE 54 1F 4A 3E 0D
    45 AE 26 C2 EF E2 A2 0E B7 FF 91 3A DA 6C ED 53
    CE C0 AF 27 3C 7E 46 61 52 FB 3D 77 9A 97 EE 23
    61 B0 1D A7 1C AF 7F 96 E0 65 D7 4E 45 7A 6C 90
    9E 73 A4 68 BF F1 D2 7B F9 11 E2 64 59 7D 0C 01
    1D 15 56 63 D1 B6 7C AF B4 BF D3 76 CA 3C DA 49
- метод офлайновой аутентификации данных CDA будет применен после выполнения первой команды GENERATE AC (пока не найдены причины, по которым этот метод не мог бы быть применен)

9. Проверка ограничений на обработку транзакции.
   - проверка соответствия номеров версий приложений карты и терминала никогда не выполняется
   - контроль использования приложения выполняется в соответствии с признаками, определенными в объекте Application Usage Control
     - терминал не является банкоматом и определено, что разрешены операции в устройствах, отличных от банкоматов
     - тип транзакции связан с покупкой товаров (услуг), код страны эмитента совпадает с кодом страны терминала и транзакции покупки товаров (услуг) внутри страны разрешены
   - контроль использования приложения показал, что нет ограничений на применение платежного приложения для выполнения транзакции
   - на карте определен объект Application Effective Date, по которому установлено, что приложение уже может использоваться
   - на карте определен объект Application Expiration Date, по которому установлено, что срок действия приложения ещё не истёк
10. Верификация владельца карты.
    - верификация владельца карты выполняется по списку CVM, в котором определены правила верификации (всего правил: 6)
    - обрабатывается правило верификации владельца карты с номером 1
      - условие выполнения: транзакция связана с выдачей наличных в банкомате
      - метод верификации: онлайновая проверка PIN-кода
      - условие выполнения метода верификации не удовлетворено (неподходящий тип транзакции или тип терминала)

- обрабатывается правило верификации владельца карты с номером 2
  - условие выполнения: терминал поддерживает метод верификации владельца карты
  - метод верификации: офлайновая проверка PIN-кода в зашифрованном виде
  - получение количества оставшихся попыток предъявления PIN-кода
    - выдача команды GET DATA для получения значения объекта платежного приложения PIN Try Counter
    - время выполнения команды: 31 мсек
    - количество оставшихся попыток предъявления PIN-кода: 3
  - предъявление зашифрованного PIN-кода карте
    - выдача команды GET CHALLENGE для получения случайного числа, используемого для шифрования PIN-кода
    - время выполнения команды: 47 мсек
    - команда вернула случайное число: 752818BC06BF70C5
    - зашифрование PIN-кода на ключе карты, используемом для шифрования PIN-кода
    - выдача команды VERIFY для проверки PIN-кода платежного приложения в зашифрованном виде
    - время выполнения команды: 281 мсек
    - предъявлен неверный PIN-код (счетчик оставшихся предъявлений равен 2)
- обрабатывается правило верификации владельца карты с номером 3
  - условие выполнения: терминал поддерживает метод верификации владельца карты
  - метод верификации: офлайновая проверка PIN-кода в открытом виде
  - получение количества оставшихся попыток предъявления PIN-кода
    - выдача команды GET DATA для получения значения объекта платежного приложения PIN Try Counter
    - время выполнения команды: 31 мсек
    - количество оставшихся попыток предъявления PIN-кода: 2
  - предъявление незашифрованного PIN-кода карте
    - выдача команды VERIFY для проверки PIN-кода платежного приложения в открытом виде
    - время выполнения команды: 63 мсек
    - предъявлен неверный PIN-код (счетчик оставшихся предъявлений равен 1)
- обрабатывается правило верификации владельца карты с номером 4
  - условие выполнения: терминал поддерживает метод верификации владельца карты
  - метод верификации: онлайновая проверка PIN-кода
  - метод верификации выполнен успешно
- обработка списка CVM завершена

11. Процедуры управления рисками, выполняемые терминалом.
    - проверка лимита платежа в офлайновом режиме:
        - сумма платежной операции: 20.00
        - максимальное значение суммы платежа в офлайновом режиме (Terminal Floor Limit): 1000.00
        - сумма платежной операции меньше максимальной суммы платежа (не обнаружена особая ситуация выполнения транзакции)
    - платежное приложение поддерживает журнал транзакций, но проверка на «split sales» никогда не выполняется в текущей версии программы
    - выполнение процедуры случайного выбора транзакции для онлайновой обработки:
        - целевой процент: 20
        - случайный процент: 53
        - пороговое значение суммы платежа для пристрастного выбора, используемое в процедуре случайного выбора транзакции для онлайновой обработки: 500.00
        - транзакция не отвечает критерию пристрастного выбора (выбор транзакции для онлайновой обработки осуществляется независимо от суммы платежной операции)
        - транзакция не выбрана для онлайновой обработки
    - проверка скорости расходования средств в офлайновом режиме не выполняется, так как на карте отсутствует объект Lower Consecutive Offline Limit
12. Оценка результатов процедур, выполненных терминалом (выработка решения о дальнейшей обработке транзакции).
    - в ходе проверок, выполненных терминалом, обнаружены следующие особые ситуации (в TVR установлены соответствующие биты):
        - online PIN entered
    - чтобы установить, требуется ли отклонение транзакции с точки зрения эквайера, используется следующий TAC-Denial: 0000000000
    - в соответствии с политикой эквайера и эмитента транзакция не отклоняется (не найдено соответствие признаков в TVR и TAC-Denial или IAC-Denial)
    - чтобы установить, требуется ли авторизация транзакции эмитентом с точки зрения эквайера, используется следующий TAC-Online: FC509C8800
        - offline data authentication was not performed
        - SDA failed
        - ICC data missing
        - card appears on terminal exception file
        - DDA failed

    - CDA failed
    - expired application
    - requested service not allowed for card product
    - cardholder verification was not successful
    - PIN required and PIN pad not present or not working
    - PIN required, PIN pad present, but PIN was not entered
    - online PIN entered
    - transaction exceeds floor limit
    - merchant forced transaction online
  - найдено соответствие следующих признаков в TVR и TAC-Online:
    - online PIN entered
  - в соответствии с политикой эквайера от платежного приложения должна быть запрошена криптограмма ARQC (авторизация транзакции в онлайновом режиме)
  - для офлайновой аутентификации данных применяется метод CDA, поэтому в первой команде GENERATE AC запрашивается сертификат Signed Dynamic Application Data

13. Выдача первой команды GENERATE AC для выполнения транзакции в контактном режиме.
  - выдается команда GENERATE AC со следующими параметрами:
    - запрашиваемая криптограмма: ARQC
    - в ответе запрашивается сертификат Signed Dynamic Application Data
  - для принятия решения о выполнении транзакции с командой передаются данные, перечисленные в CDOL1:
    00 00 00 00 20 00 00 00 00 00 00 00 06 43 00 00
    04 00 00 06 43 19 02 12 00 DC 6E 0B 1C 22 00 00
    00 00 00 00 00 00 00 00 02 03 00
  - интерпретация данных в соответствии с CDOL1:
    - Amount, Authorized (numeric) (9F02.6): 20.00
    - Amount, Other (numeric) (9F03.6): 0.00
    - Terminal Country Code (9F1A.2): 0643
    - Terminal Verification Results (95.5): 0000040000
      - online PIN entered
    - Transaction Currency Code (5F2A.2): 0643
    - Transaction Date (9A.3): 12.02.2019
    - Transaction Type (9C.1): 00 (покупка товаров или услуг)
    - Unpredictable Number (9F37.4): DC6E0B1C

    - Terminal Type (9F35.1): 22
        - Attended, Offline with online capability
        - Operational control provided by Merchant
    - Data Authentication Code (DAC) (9F45.2): 0000
    - ICC Dynamic Number (9F4C.8): 0000000000000000
    - CVM Results (9F34.3): 020300
        - Enciphered PIN verified online
        - If terminal supports the CVM
        - Unknown CVM Result
- время выполнения команды: 344 мсек
- в ответ на команду получены следующие данные:

  77 81 A2 9F 27 01 80 9F 36 02 00 39 9F 4B 81 80
  11 7B CB 74 AF 63 52 12 4B 99 E9 54 C6 DB 9E 67
  24 3C 7B 49 F6 E5 A5 2D 0D E5 F1 5F 47 5C 54 0B
  DC FF C6 26 64 F7 D8 B1 90 38 54 A0 9B B2 F3 2D
  87 9B F6 51 84 5A 2D C1 9E 63 75 81 E0 41 F4 50
  D6 86 4A A0 C5 A0 05 7D B4 16 82 1C 2E B8 43
  A7 1F 5C 44 56 88 D6 C9 5A 5B B9 11 B2 3D CE 05
  20 40 C7 B8 89 35 70 54 7B 5C 12 37 71 5D C9 C8
  EE 6C 0B CF 41 B0 A4 7D 26 70 07 75 1C D5 3B 00
  9F 10 12 01 10 A4 40 01 12 00 00 00 00 00 00 00
  04 20 00 00 FF
- интерпретация полученной TLV-структуры:
    - 77.162 Response Message Template Format 2
        - 9F27.1 Cryptogram Information Data (CID)
        - 9F36.2 Application Transaction Counter (ATC)
        - 9F4B.128 Signed Dynamic Application Data
        - 9F10.18 Issuer Application Data
- выполняется анализ данных, полученных в ответ на команду
    - Cryptogram Information Data: 80
        - ARQC (Authorisation Request Cryptogram - Online authorisation requested)
    - ATC: 0039

- Signed Dynamic Application Data:
  11 7B CB 74 AF 63 52 12 4B 99 E9 54 C6 DB 9E 67
  24 3C 7B 49 F6 E5 A5 2D 0D E5 F1 5F 47 5C 54 0B
  DC FF C6 26 64 F7 D8 B1 90 38 54 A0 9B B2 F3 2D
  87 9B F6 51 84 5A 2D C1 9E 63 75 81 E0 41 F4 50
  D6 86 4A A0 C5 A0 A0 05 7D B4 16 82 1C 2E B8 43
  A7 1F 5C 44 56 88 D6 C9 5A 5B B9 11 B2 3D CE 05
  20 40 C7 B8 89 35 70 54 7B 5C 12 37 71 5D C9 C8
  EE 6C 0B CF 41 B0 A4 7D 26 70 07 75 1C D5 3B 00
- Issuer Application Data: 0110A44001120000000000000004200000FF
  - Derivation key index: 1
  - Cryptogram Version Number: 16
  - Card Verification Results:
    - AC returned in First Generate AC: ARQC
    - AC returned in Second Generate AC: second Generate AC not requested
    - Offline PIN verification performed
    - CDA returned in First Generate AC
    - Script Counter: 0
    - PIN Try Counter: 1
    - Offline PIN verification failed
    - Domestic transaction
  - DAC/ICC Dynamic Number 2 Bytes: 0000
  - Counters: 00000004200000FF
- в данных, полученных в ответ на команду, ошибки не обнаружены
- проверяется сертификат Signed Dynamic Application Data
- сертификат Signed Dynamic Application Data признан достоверным
- в сертификате определен следующий ICC Dynamic Number: 5CA0B7A2ED4ABEB0
- из сертификата извлечена криптограмма платежного приложения: 1CECDF76E8151DD9
- метод офлайновой аутентификации данных CDA выполнен успешно

14. Проверка криптограммы платежного приложения, предоставленной первой командой GENERATE AC, с использованием данных приложения и заданного значения ключа для вычисления криптограмм.
    - проверка криптограммы платежного приложения с данным RID не реализована

15. Онлайновая обработка (эмуляция действий терминала в случае, когда транзакция должна быть отправлена на авторизацию эмитенту).
   - ситуация, которая по требованию пользователя должна быть сымитирована в процессе онлайновой обработки: терминал запрашивает одобрение транзакции (с эмуляцией состояния «Unable to go Online»)
   - чтобы установить, требуется ли отклонение транзакции с точки зрения эквайера в состоянии «Unable to go Online», используется следующий TAC-Default: FC509C8800
     - offline data authentication was not performed
     - SDA failed
     - ICC data missing
     - card appears on terminal exception file
     - DDA failed
     - CDA failed
     - expired application
     - requested service not allowed for card product
     - cardholder verification was not successful
     - PIN required and PIN pad not present or not working
     - PIN required, PIN pad present, but PIN was not entered
     - online PIN entered
     - transaction exceeds floor limit
     - merchant forced transaction online
   - найдено соответствие следующих признаков в TVR и TAC-Default:
     - online PIN entered
   - в соответствии с политикой эквайера транзакция должна быть отклонена
16. Выдача второй команды GENERATE AC для принятия окончательного решения об обработке транзакции после онлайновой обработки.
   - выдается команда GENERATE AC со следующими параметрами:
     - запрашиваемая криптограмма: AAC
     - в ответе не запрашивается сертификат Signed Dynamic Application Data
   - для принятия решения о выполнении транзакции с командой передаются данные, перечисленные в CDOL2:
     00 00 00 00 00 00 00 00 00 00 5A 33 00 00 04 00
     00 DC 6E 0B 1C 5C A0 B7 A2 ED 4A BE B0
   - интерпретация данных в соответствии с CDOL2:
     - Issuer Authentication Data (91.10): 00000000000000000000

        - Issuer Authentication Data not received by terminal
    - Authorization Response Code (8A.2): 'Z3'
    - Terminal Verification Results (95.5): 0000040000
        - online PIN entered
    - Unpredictable Number (9F37.4): DC6E0B1C
    - ICC Dynamic Number (9F4C.8): 5CA0B7A2ED4ABEB0
- время выполнения команды: 172 мсек
- в ответ на команду получены следующие данные:
  77 29 9F 27 01 00 9F 36 02 00 39 9F 26 08 5C 96
  26 33 1B 95 C9 B4 9F 10 12 01 10 24 40 01 52 00
  00 5C A0 00 00 00 04 20 00 00 FF
- интерпретация полученной TLV-структуры:
    - 77.41 Response Message Template Format 2
        - 9F27.1 Cryptogram Information Data (CID)
        - 9F36.2 Application Transaction Counter (ATC)
        - 9F26.8 Application Cryptogram
        - 9F10.18 Issuer Application Data
- выполняется анализ данных, полученных в ответ на команду
    - Cryptogram Information Data: 00
        - AAC (Application Authentication Cryptogram - Transaction declined)
    - ATC: 0039
    - Application Cryptogram: 5C9626331B95C9B4
    - Issuer Application Data: 0110244001520005CA000000004200000FF
        - Derivation key index: 1
        - Cryptogram Version Number: 16
        - Card Verification Results:
            - AC returned in First Generate AC: ARQC
            - AC returned in Second Generate AC: AAC
            - Offline PIN verification performed
            - CDA returned in First Generate AC
            - Script Counter: 0
            - PIN Try Counter: 1
            - Unable to go online

            - Offline PIN verification failed
            - Domestic transaction
        - DAC/ICC Dynamic Number 2 Bytes: 5CA0
        - Counters: 00000004200000FF
    - в данных, полученных в ответ на команду, ошибки не обнаружены

17. Проверка криптограммы платежного приложения, предоставленной второй командой GENERATE AC, с использованием данных приложения и заданного значения ключа для вычисления криптограмм.
    - проверка криптограммы платежного приложения с данным RID не реализована

Проверка платежной карты в контактном режиме завершена, поскольку все требуемые операции с картой выполнены. В процессе проверки были выполнены следующие действия:

- платежное приложение выбрано на карте с помощью команды SELECT
- выдана команда GET PROCESSING OPTIONS для инициирования транзакции и получения информации, необходимой для её выполнения
- считаны данные из записей файлов платежного приложения
- восстановлен публичный ключ эмитента
- восстановлен публичный ключ карты
- успешно выполнена офлайновая аутентификация данных карты
- выполнен метод верификации владельца карты «Предъявление PIN-кода для передачи его эмитенту»
- выданы команды GET DATA для получения информации об объектах платежного приложения
- выдана первая команда GENERATE AC для выполнения транзакции в контактном режиме
- выдана вторая команда GENERATE AC для завершения транзакции в контактном режиме

Дополнительная документация

При чтении документации на комплекс тестирования ECV может потребоваться дополнительное изучение следующих спецификаций и стандартов.

1. EMV. Integrated Circuit Card Specifications for Payment Systems. Book 1. Application Independent ICC to Terminal Interface Requirements. Version 4.2. June 2008.

2. EMV. Integrated Circuit Card Specifications for Payment Systems. Book 2. Security and Key Management. Version 4.2. June 2008.

3. EMV. Integrated Circuit Card Specifications for Payment Systems. Book 3. Application Specification. Version 4.2. June 2008.

4. EMV. Integrated Circuit Card Specifications for Payment Systems. Book 4. Cardholder, Attendant, and Acquirer Interface Requirements. Version 4.2. June 2008.

5. EMV. Contactless Specifications for Payment Systems. Book A. Architecture and General Requirements. Version 2.1. March 2011.

6. EMV. Contactless Specifications for Payment Systems. Book B. Entry Point Specification. Version 2.1. March 2011.

7. EMV Card Personalization Specification. Version 1.1. July 2007.

8. ISO/IEC 7816-4. Identification Cards – Integrated Circuit Cards. Part 4. Organization, security and commands for interchange.

9. ISO/IEC 14443-1. Identification Cards – Contactless Integrated Circuit Cards – Proximity Cards. Part 1. Physical characteristics. 2008.

10. ISO/IEC 14443-2. Identification Cards – Contactless Integrated Circuit Cards – Proximity Cards. Part 2. Radio frequency power and signal interface. 2010.

11. ISO/IEC 14443-3. Identification Cards – Contactless Integrated Circuit Cards – Proximity Cards. Part 3. Initialization and anti-collision. 2011.

12. ISO/IEC 14443-4. Identification Cards – Contactless Integrated Circuit Cards – Proximity Cards. Part 4. Transmission protocol. 2008.

13. ISO/IEC 8825-1. Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER). 2008.

14. ANSI X9.24-2009. Retail Financial Services Symmetric Key Management. Part 1: Using Symmetric Techniques. 2009.

15. ISO 9564-1. Financial services – Personal Identification Number (PIN) management and security. Part 1: Basic principles and requirements for PINs in card-based systems. 2011.

Если вы хотите более глубоко изучить проблемы, связанные с безналичными платежами, платежными картами, спецификациями EMV, и еще узнать много интересного, рекомендуется прочесть книгу Игоря Михайловича Голдовского «Банковские микропроцессорные карты». ISBN 978-5-9614-1233-8; 2010 г.

Д Л Я   З А М Е Т О К